Lecture Notes on Testing Graph Properties in the Dense Graph Model

Oded Goldreich[*]

February 24, 2016

Summary: Following a general introduction to testing graph properties, this lecture focuses on the dense graph model, where graphs are represented by their adjacency matrix (predicate). The highlights of this lecture include:

1. A presentation of a natural class of graph properties that can each be tested within query complexity that is polynomial in the reciprocal of the proximity parameter. This class, called general graph partition problems, contains properties such as k-`Colorability` (for any $k \geq 2$) and properties that refer to the density of the max-clique and to the density of the max-cut in a graph.
2. An exposition of the connection of testing (in this model) to Szemeŕedi's Regularity Lemma. The starting point and pivot of this exposition is the existence of constant-query (one-sided error) proximity-oblivious testers for all subgraph freeness properties.

We conclude this lecture with a taxonomy of known testers, organized according to their query complexity.

The current notes are based on many sources; see Section 6.1 for details. With the exception of Section 4, the text was adapted from [24] (and extensively revised to fit its current use).

Organization. The current lecture is the first out of a series of three lectures that cover three models for testing graph properties. In each model, we spell out the definition of property testing (when specialized to that model), present some of the known results, and demonstrate some of the ideas involved (by focusing on testing `Bipartiteness`, which seems a good benchmark).

We start the current lecture with a general introduction to testing graph properties, which includes an overview of the three models (see Section 1.2). We then present and illustrate the "dense graph model" (Section 2), which is the focus of the current lecture. The main two sections (i.e., Sections 3 and 4) cover the two topics that are mentioned in the foregoing summary: Section 3 deals with testing arbitrary graph partition properties, as illustrated by the example of testing `Bipartitness`. Section 4 deals with the connection between property testing in this model and Szemeŕedi's Regularity Lemma, as illustrated by testing subgraph-freeness. The last two sections (i.e., Sections 5 and 6) are descriptive in nature: Section 5 presents a taxonomy of the known results, whereas Section 6 presents final comments.

[*]Department of Computer Science, Weizmann Institute of Science, Rehovot, ISRAEL.

1 The general context: Introduction to testing graph properties

> *Graph theory has long become recognized as one of the more useful mathematical subjects for the computer science student to master. The approach which is natural in computer science is the algorithmic one; our interest is not so much in existence proofs or enumeration techniques, as it is in finding efficient algorithms for solving relevant problems, or alternatively showing evidence that no such algorithms exist. Although algorithmic graph theory was started by Euler, if not earlier, its development in the last ten years has been dramatic and revolutionary.*
>
> Shimon Even, *Graph Algorithms*, 1979.

Meditating on these facts, one may ask what is the source of this ubiquitous use of graphs in computer science. The most common answer is that graphs arise naturally as a model (or an abstraction) of numerous natural and artificial objects. Another answer is that graphs help visualize binary relations over finite sets. These two different answers correspond to two types of models of testing graph properties that will be discussed below. But before doing so, let us recall some basic background.

Teaching note: We believe that most readers can afford skipping Section 1.1, which presents the basic notions and terminology regarding graphs. The vocabulary includes terms such as vertex, edge, simple graph, incident, adjacent, degree, path, cycle, subgraph, induced graph, and isomorphism between graphs.

1.1 Basic background

A simple graph $G=(V,E)$ consists of a *finite* set of vertices V and a finite set of edges E, where each edge is an *unordered pair* of vertices; that is, $E \subseteq \binom{V}{2} \stackrel{\rm def}{=} \{\{u,v\} : u,v \in V \wedge u \neq v\}$. This formalism does not allow self-loops and parallel edges, which are allowed in general (i.e., non-simple) graphs, where E is a multi-set that may contain (in addition to two-element subsets of V also) singletons (i.e., self-loops). The vertex u is called an end-point of the edge $\{u,v\}$, and the edge $\{u,v\}$ is said to be incident at v. In such a case we say that u and v are adjacent in the graph, and that u is a neighbor of v. The degree of a vertex in G is defined as the number of edges that are incident at this vertex.

We will consider various sub-structures of graphs, the simplest one being paths. A path in a graph $G=(V,E)$ is a sequence of vertices $(v_0,...,v_\ell)$ such that for every $i \in [\ell] \stackrel{\rm def}{=} \{1,...,\ell\}$ it holds that v_{i-1} and v_i are adjacent in G. Such a path is said to have length ℓ. A simple path is a path in which each vertex appears at most once, which implies that the longest possible simple path in G has length $|V|-1$. The graph is called connected if there exists a path between each pair of vertices in it.

A cycle is a path in which the last vertex equals the first one (i.e., $v_\ell = v_0$). The cycle $(v_0,...,v_\ell)$ is called simple if $\ell > 2$ and $|\{v_0,...,v_\ell\}| = \ell$ (i.e., if $v_i = v_j$ then $i \equiv j \pmod{\ell}$, and the cycle (u,v,u) is not considered simple). A graph is called acyclic (or cycle-free or a forest) if it has no simple cycles, and if it is also connected then it is called a tree. Note that $G=(V,E)$ is a tree if and only if it is connected and $|E| = |V|-1$, and that there is a unique simple path between each pair of vertices in a tree.

A subgraph of the graph $G=(V,E)$ is any graph $G'=(V',E')$ satisfying $V' \subseteq V$ and $E' \subseteq E$. Note that a simple cycle in G is a connected subgraph of G in which each vertex has degree exactly two. An induced subgraph of the graph $G=(V,E)$ is any subgraph $G'=(V',E')$ that contains all edges of E that have both endpoints in V'. In such a case, we say that G' is the subgraph induced by V'.

Two graphs, $G_1=(V_1,E_1)$ and $G_1=(V_2,E_2)$ are said to be isomorphic if there exists a bijection $\phi : V_1 \to V_2$ such that $E_2 = \{\{\phi(u),\phi(v)\} : \{u,v\} \in E_1\}$; that is, $\phi(u)$ is adjacent to $\phi(v)$ in G_2 if and only if u is adjacent to v in G_1.

1.2 Three Models of Testing Graph Properties

The fact that we call the objects of our study "graphs" is meaningless unless our study refers to characteristics of these objects, which may not be shared by other objects. What distinguishes the edge set E of a graph $G=(V,E)$ from any other set of similar cardinality is that we can refer to it via V; that is, E is an adjacency relation over V, and so the existence of edges $\{u,v_1\}$ and $\{u,v_2\}$ that share a common end-point is different from the existence of two other edges that do not share an end-point. A cycle of length t is not an arbitrary sequence of t elements of E, but rather one with a specific structure. Furthermore, we are interested in properties that are invariant under renaming of the vertices. Such properties are called *graph properties.*

Definition 1 (graph properties): *A* graph property *is a set of graphs that is closed under graph isomorphism. That is,* Π *is a graph property if, for every graph* $G=(V,E)$ *and every bijection* $\pi : V \to V'$*, it holds that* $G \in \Pi$ *if and only if* $\pi(G) \in \Pi$*, where* $\pi(G)$ *is the graph obtained from* G *by relabelling the vertices according to* π*; that is,*

$$\pi(G) \stackrel{\rm def}{=} (V, \{\{\pi(u),\pi(v)\} : \{u,v\} \in E\}).$$

For sake of simplicity, we shall consider only graphs $G=(V,E)$ with vertex set $V = \{1,...,|V|\}$. (Wishing to reserve n for the size of the representatioon of the tested object, we shall often denote the number of vertices by $k=|V|$.)

In light of what we have seen so far, a tester for a graph property Π is a randomized algorithm that is given oracle access to a graph, $G=(V,E)$, and has to determine whether the graph is in Π or is far from being in Π. But the foregoing falls short from constituting a sound definition. We have to specify what does it mean to be given oracle access to a graph, and when are two graphs considered to be far from one another. That is, we have to specify the meaning of "oracle access to a graph" (i.e., the type of queries that are allowed to the graph) as well as the distance-measure (between pairs of graphs). Recall that, as stated in the first lecture, these (pairs of) choices are of key importance. There are at least three natural (pairs of) choices, and each of them yields a different model. Three such models are reviewed next.

The dense graphs (a.k.a adjacency predicate) model. Here the graph $G=(V,E)$ is represented by the *adjacency predicate* $g : \binom{V}{2} \to \{0,1\}$ such that $\{u,v\} \in E$ if and only if $g(\{u,v\}) = 1$. Hence, oracle access to G means oracle access to g, and the distance between graphs (with vertex set V) is defined as the distance between their corresponding representations (which have size $\binom{|V|}{2}$); that is, if the graphs G and G' are represented by the functions g and g', then their relative distance is the fraction of pairs $\{u,v\}$ such that $g(\{u,v\}) \neq g'(\{u,v\})$ (i.e., $|\{\{u,v\} : g(\{u,v\}) \neq g'(\{u,v\})\}|/\binom{|V|}{2}$).

It will be more convenient to represent the graph $G = (V, E)$ by the symmetric function $g : V \times V \to \{0, 1\}$ such that is $g(u, v) = 1$ if and only if $\{u, v\} \in E$. This representation is slightly redundant, since $g(u, v) = g(v, u)$ and $g(v, v) = 0$ always holds, but it is less cumbersome.[1]

Note that saying that $G = (V, E)$ is ϵ-far from the graph property Π means that for every $G' \in \Pi$ it holds that G is ϵ-far from G'. Since Π is closed under graph isomorphism, this means that G is ϵ-far from any isomorphic copy of G'; that is, for every permutation π over V, it holds that $|\{(u, v) : g(u, v) \neq g'(\pi(u), \pi(v))\}| > \epsilon \cdot |V|^2$, where $g : V^2 \to \{0, 1\}$ and $g' : V^2 \to \{0, 1\}$ are as above.

Finally, note that this notion of distance between graphs is most meaningful in the case that the graphs are dense (since in this case the fraction of the number of possible vertex pairs is closely related to the fraction of the actual number of edges). Thus, this model is often called the *dense graph model*.

The bounded-degree graph (a.k.a incidence function) model. Here, for some fixed upper bound d (on the degrees of vertices in G), the graph $G = (V, E)$ is represented by the *incidence function* $g : V \times [d] \to V \cup \{\perp\}$ such that $g(u, i) = v$ if v is the i^{th} vertex incident at u and $g(u, i) = \perp$ if u has less than i neighbors. (Indeed, this representation assumes and/or induces an order on the neighbors of each vertex in G, and this representation is redundant since each edge is represented twice.)[2] As before, oracle access to G means oracle access to g, but g is different here. Likewise, the distance between graphs (with vertex set V) is defined as the distance between their corresponding representations (which have size $|V| \cdot d$); that is, if the graphs G and G' are represented by the functions g and g', then their relative distance is the fraction of pairs (u, i) such that $g(u, i) \neq g'(u, i)$.

Indeed, only graphs of degree at most d can be represented in this model, which is called the *bounded-degree graph model*.

Again, saying that $G = (V, E)$ is ϵ-far from the graph property Π means that for every $G' \in \Pi$ it holds that G is ϵ-far from G'. Since Π is closed under graph isomorphism and the ordering of the vertices incident at each vertex is arbitrary, this means that for every permutation π over V, it holds that

$$\sum_{u \in V} |\{v : \exists i\ g(u, i) = v\} \triangle \{v : \exists i\ g'(\pi(u), i) = \pi(v)\}| \ > \ \epsilon d N \,,$$

where g and g' are as above, and $\triangle$ denotes the symmetric difference (i.e., $A \triangle B = (A \cup B) \setminus (A \cap B)$).

Note that, both in the dense graph model and in the bounded-degree graph model, the (relative) distance between graphs is measured according to the representation of these graphs as functions, but the representation is different in the two models, and so the (relative) distances are different in the two models. In contrast to the foregoing two models in which the oracle queries and the (relative) distances between graphs are linked to the representation of graphs as functions, in the following model the representation is blurred and the query types and distance measure are decoupled.

[1] Note that representing G and G' by $g : V \times V \to \{0, 1\}$ and $g' : V \times V \to \{0, 1\}$, means that the relative distance between g and g' is $|\{(u, v) : g(u, v) \neq g'(\pi(u), \pi(v))\}|/|V|^2$.

[2] That is, we always assume that $g(u, i) = v$ if and only if there exists a $j \in [d]$ such that $g(v, j) = u$. We stress that j does not necessarily equal i.

The general graph model. Here the graphs are redundantly represented by both their adjacency predicate and their incidence functions (while not assuming a degree bound (except for the obvious bound of $|V|-1$)), but this representation is implicit in the type of queries allowed (i.e., the algorithm can make queries of both types) and does not effect the distance measure. Instead, the relative distance between the graphs $G=(V,E)$ and $G'=(V,E')$ is defined as $\frac{|E\triangle E'|}{\max(|E|,|E'|)}$; that is, the absolute distance is normalized by the actual number of edges rather than by an absolute upper bound (on the number of edges) such as $\binom{|V|}{2}$ or $d|V|/2$.

Needless to say, the general graph model is the most general one, and it is indeed closest to actual algorithmic applications. In other words, this model is relevant for most applications, since these seem to refer to general graphs (which model various natural and artificial objects). In contrast, the dense graph model is relevant to applications that refer to (dense) binary relations over finite sets, whereas the bounded-degree graph model is relevant only to applications in which the vertex degree is bounded.

The fact that the *general graph model* has received relatively little attention (so far) merely reflects the fact that its study is overly complex. Given that current studies of the other models still face formidable difficulties (and that these models offer a host of interesting open problems), it is natural that researchers shy away from yet another level of complication.

> **Teaching note:** While the following comment applies to property testing at large, it seems appropriate to make it (and stress it) in the context of testing graph properties, since this context seems closest to standard algorithmic research.

The current focus on query complexity. Although property testing is motivated by referring to super-fast algorithms, research in the area tends to focus on the *query complexity* of testing various properties. This focus should be viewed as providing an initial estimate to the actual complexity of the testing problems involved; certainly, query-complexity lower bounds imply corresponding bounds on the time complexity, whereas the latter is typically at most exponential in the query complexity. Furthermore, in many cases, the time complexity is polynomial in the query complexity and this fact is typically stated. Thus, we will follow the practice of focusing on the query complexity of testing, but also mention time complexity upper bounds whenever they are of interest.

Digest: The issue of representation in light of the three models. As stated in the first lecture, the distinction between objects and their representation is typically blurred in computer science; nevertheless, this distinction is important. Indeed, reasonable and/or natural representations are always assumed either explicitly or implicitly (see, e.g., [23, Sec. 1.2.1]). The specific choice of a reasonable and/or natural representation becomes crucial when one considers the exact complexity of algorithms (as is common in algorithmic research), rather than their general "ball park" (e.g., being in the complexity class $\mathcal{P}$ or not).

The representation is even more crucial in our context (i.e., in the study of property testing). This is the case for two reasons, which transcend the standard algorithmic concerns:

1. We are interested in sub-linear time algorithms, which means that these algorithms query bits in the representation of the object. Needless to say, different representations mean different types of queries, and this difference is crucial when one does not fully recover the object by queries.

2. We are interested in the distance between objects (or, actually, in the distance between objects and sets of objects), whereas this distance may be measured in terms of the distance between their representations. In such a case, different representations of objects may yield vastly different distances between the same objects.

In light of the above, when considering property testing, we always detail the exact representation of the objects. The three foregoing models use different representations of the same objects, which means that the algorithms in the different models have different query capacities and their performance is evaluated with respect to different distance measures. We believe that the types of queries allowed in each model constitute the natural choice for that model. In the first two models, the underlying representation also provides a natural basis for the definition of a distance measure between objects, whereas in the third model the definition of the distance measure is decoupled from the representation of the objects (and refers to their "actual size").

2 The Dense Graph Model: Some basics

In this section we spell out the actual definition of "testing graph properties in the dense graph model" (Section 2.1) and outline a couple of simple testers, which in some sense are based on artifacts of this specific model (Section 2.2). In contrast, in Section 2.3, we illustrate how the fact that we deal with graphs complicates the analysis of a seemingly simple tester.

2.1 The actual definition

In the adjacency matrix model (a.k.a the dense graph model), an k-vertex graph $G = ([k], E)$ is represented by the Boolean function $g : [k] \times [k] \to \{0, 1\}$ such that $g(u, v) = 1$ if and only if u and v are adjacent in G (i.e., $\{u, v\} \in E$). Distance between graphs is measured in terms of their aforementioned representation (i.e., as the fraction of (the number of) different matrix entries (over k^2)), but occasionally one uses the more intuitive notion of the fraction of (the number of) unordered vertex pairs over $\binom{k}{2}$.[3]

Recall that we are interested in *graph properties*, which are sets of graphs that are closed under isomorphism; that is, Π is a graph property if for every graph $G = ([k], E)$ and every permutation π of $[k]$ it holds that $G \in \Pi$ if and only if $\pi(G) \in \Pi$, where $\pi(G) \stackrel{\text{def}}{=} ([k], \{\{\pi(u), \pi(v)\} : \{u, v\} \in E\})$. We now spell out the meaning of property testing in this model.[4]

Definition 2 (testing graph properties in the adjacency matrix model): *A* tester *for a graph property Π is a probabilistic oracle machine that, on input parameters k and ϵ and access to* (the adjacency predicate of) *an k-vertex graph $G = ([k], E)$, outputs a binary verdict that satisfies the following two conditions.*

1. If $G \in \Pi$, then the tester accepts with probability at least $2/3$.

[3] Indeed, there is a tiny discrepancy between these two measures, but it is immaterial in all discussions. Note that, for sake of technical convenience, we chose to use a redundant representation (i.e., $g(u, v) = g(v, u)$ and $g(v, v) = 0$), and that we denote the number of vertices by k in order to maintain with the convention that n denotes the size of the representation (i.e., $n = k^2$).

[4] Indeed, we slightly deviate from the conventions of the first lecture by providing the tester with k (which denotes the number of vertices in G) rather than with $n = k^2$ (which denotes the size of the domain of the function g).

2. *If G is ϵ-far from Π, then the tester accepts with probability at most $1/3$, where G is* ϵ-far *from Π if for every k-vertex graph $G' = ([k], E') \in \Pi$ it holds that the symmetric difference between E and E' has cardinality that is greater than $\epsilon \cdot k^2/2$* (equiv., the representations of G and G' as adjacency predicates differ on more than $\epsilon \cdot k^2$ vertex-pairs).[5]

If the tester accepts every graph in Π with probability 1, then we say that it has one-sided error*; otherwise, we say that it has* two-sided error*. A tester is called* non-adaptive *if it determines all its queries based solely on its internal coin tosses* (and the parameters k and ϵ)*; otherwise, it is called* adaptive.

The query complexity of a tester is the number of queries it makes to any k-vertex graph, as a function of the parameters k and ϵ.[6] We say that a tester is efficient if it runs in time that is linear in its query complexity, where basic operations on elements of $[k]$ (and in particular, uniformly selecting an element in $[k]$) are counted at unit cost.

We stress that testers are defined as (uniform) algorithms that are given the size parameter k and the distance (or proximity) parameter ϵ as explicit inputs.[7] This uniformity (over the values of the distance parameter) makes the positive results stronger and more appealing (especially in light of a separation result shown in [9]). In contrast, negative results typically refer to a fixed value of the distance parameter.

Representing graphs by their adjacency predicate is very natural, but it is quite problematic if the input graph is not dense (i.e., if $|E| = o(k^2)$). In such a case (i.e., when G is not dense), queries to the oracle are likely to be uninformative (e.g., a uniformly distributed query is answered by 0 with probability $1 - o(1)$). On the other hand, each non-dense graph is $o(1)$-close to the empty graph, so if the latter has the property (and we are guaranteed that the tested graph is non-dense), then testing is trivial (for any constant $\epsilon > 0$). All these reservations are not applicable when the tested graph is dense, as is the case when the graph is used to represent a (symmetric) binary relation that is satisfied quite frequently (say, with constant frequency).

2.2 Abuses of the model: Trivial and sparse properties

In continuation to the foregoing discussion, we note that graph properties can be trivial to test also when the input graph is dense. One such case is when every k-vertex graph is ϵ-close to the property (for every $\epsilon > k^{-\Omega(1)}$). This is the case with many natural graph properties: for example, every k-vertex graph is $O(1/k)$-close to being connected (or even Hamiltonian and Eulerian), and ditto with respect to being unconnected.

Proposition 3 (trivially testable properties (in the dense graph model)): *Let Π be a graph property and $c > 0$. If every k-vertex graph is k^{-c}-close to Π, then ϵ-testing Π with one-sided error can be done with zero queries if $\epsilon \geq k^{-c}$ and with $(1/\epsilon)^{2/c}$ queries otherwise.*

[5] Indeed, it is more natural to consider the symmetric difference between E and E' as a fraction of $\binom{k}{2}$, but it is more convenient to adopt the alternative normalization.

[6] As in Footnote 4, we deviated from the convention of presenting the query complexity as a function of $n = k^2$ and ϵ.

[7] That is, we refer to the standard (uniform) model of computation (cf., e.g., [23, Sec. 1.2.3]), which does not allow for hard-wiring of some parameters (e.g., input length) into the computing device (as done in the case of non-uniform circuit families).

Proof: If $\epsilon \geq k^{-c}$, then the tester accepts the graph without making any query (since, in this case, the graph is ϵ-close to Π). Otherwise (i.e., $\epsilon < k^{-c}$), the tester just retrieves the entire graph and decides accordingly, but in this case $k^2 < (1/\epsilon)^{2/c}$. ■

Another case when testing is easy, alas not that trivial, is when the property is satisfied only for sparse graphs. For example, consider being *planar* or being *cycle-free*.[8] In such a case, testing the property reduces to checking that the graph is sparse enough and retrieving it only in that case.

Proposition 4 (testing "sparse graph" properties in the dense graph model): *Let Π be a graph property and $c < 2$. If every k-vertex graph in Π has at most k^c edges, then ϵ-testing Π can be done in* $\mathrm{poly}(1/\epsilon)$ *many queries. In particular, if $\epsilon \geq 3k^{-(2-c)}$ then $O(1/\epsilon)$ queries suffice.*

(Note that this tester has two-sided error.)

Proof: If $\epsilon \geq 3k^{-(2-c)}$, then the tester uses $O(1/\epsilon)$ random queries to estimate the edge density of the graph such that it distinguishes between density at least $2\epsilon/3$ and density at most $\epsilon/3$.[9] In the first case the tester rejects (since the graph is far enough from being sufficiently sparse), and in the second case the tester accepts (since the graph is close enough to the empty graph, which is close enough to Π). Otherwise (i.e., when $\epsilon < 3k^{-(2-c)}$), the tester just retrieves the entire graph and decides accordingly, but in this case $k^2 < (3/\epsilon)^{2/(2-c)}$. ■

2.3 Testing degree regularity

A case in which the the fact that we deal with graphs actually makes life harder is that of testing *degree regularity*. A graph is called regular if all its vertices have the same degree; that is, $G = ([k], E)$ is regular if there exists an integer d such that $d_G(u) \stackrel{\text{def}}{=} |\{v : \{u, v\}|$ equals d for every $u \in [k]$. In such a case we say that G is d-regular.

Theorem 5 (testing degree regularity in the dense graph model): *Degree regularity can be tested by using $O(1/\epsilon^2)$ non-adaptive queries.*

We note that this upper bound is tight (see Exercise 1). As discussed in the proof (see Claim 5.1), the tester is identical to one that could be used to test that a k-by-k Boolean matrix has rows of equal Hamming weight, but its analysis is more complex in the current setting (in which the matrix must be symmetric and lack 1-entries on its diagonal). The point is that it is not obvious that if the average deviation of the degrees of vertices in the graph (from some value) is small, then the graph is close to being regular. (In contrast, it is obvious that if the average deviation of the weights of rows in a matrix (from some value) is small, then the matrix is close to having equal weight rows.)

Proof: We start by reviewing a simpler tester of query complexity $\widetilde{O}(1/\epsilon^3)$. This tester selects $O(1/\epsilon)$ random vertices, and estimates the degree of each of them up to $\pm 0.01\epsilon k$ using a sample of $s = \widetilde{O}(1/\epsilon^2)$ random vertices (and making the corresponding s queries).[10] The tester accepts if and only if all these estimates are at most $0.02\epsilon k$ apart.

[8] Recall that any k-vertex planar graph has at most $\max(k-1, 3k-6)$ edges, whereas any (k-vertex) cycle-free graph has at most $k-1$ edges.

[9] The analysis uses a multiplicative Chernoff bound.

[10] Recall that we can the estimate of the average value of a function $f : [k] \to \{0, ..., k-1\}$ by a sample of size $O(t/\epsilon^2)$ such that, with probability at least $1 - 2^{-t}$, the estimate is within an additive deviation of $0.01\epsilon k$ from the actual value.

If G is regular, then the tester will accept it with high probability. On the other hand, if the tester accepts G with high probability, then we can infer that there exists an integer d such that all but at most $0.02\epsilon k$ of the vertices have degree $d\pm(0.02\epsilon k+1)$. (This can be shown by considering the $0.01\epsilon k$ vertices of highest degree and $0.01\epsilon k$ vertices of lowest degree.)[11] The analysis is completed by proving that in this case the graph G is ϵ-close to regular.

Claim 5.1 (local-vs-global distance to degree regularity): *If $d < k$ and $dk/2$ are natural numbers and $\sum_{v\in[k]} |d_G(v) - d| \leq \epsilon' \cdot k^2$, then G is $6\epsilon'$-close to the set of d-regular k-vertex graphs.*

(Indeed, $\sum_{v\in[k]} |d_G(v) - d|$ represents the "local" distance of G from being regular, whereas we are interested in the "global" distance as captured by Definition 2.) Note that a version of Claim 5.1 that refers to a k-by-k Boolean matrix G and lets $d_G(v)$ denote the Hamming weight of row v is trivial. In that case (of general Boolean matrices), the matrix G is ϵ'-close to a matrix in which all rows have weight d. But the latter matrix is not necessarily symmetric and may have 1-entries on the diagonal (i.e., it does not necessarily correspond to an adjacency matrix of a graph). Turning back to our application, note if there exists an integer d such that all but at most $0.02\epsilon k$ of the vertices in the graph G have degree $d\pm(0.03\epsilon k)$, then $\sum_{v\in[k]} |d_G(v)-d| < 0.02\epsilon k\cdot(k-1)+k\cdot 0.03\epsilon k < 0.05\epsilon k^2$. (This assumes that dk is even; otherwise we can use $d-1$ instead of d.)[12]

Proof: We modify G in three stages, while keeping track of the number of edge modifications. In the first stage we reduce all vertex degrees to at most d, by scanning all vertices and omitting $d_G(v)-d$ edges incident at each vertex $v \in H \stackrel{\rm def}{=} \{u : d_G(u) > d\}$. Since $\sum_{v\in H}(d_G(v)-d) \leq \epsilon' k^2$, we obtain a graph G' that is $\frac{\epsilon' k^2}{k^2/2}$-close to G such that $d_{G'}(v) \leq d$ holds for each vertex v, because every omitted edge reduces $\sum_{v\in H}\max(0, d_G(v)-d)$ by at least one unit. Furthermore, $\sum_{v\in[k]} |d_{G'}(v)-d| \leq \epsilon'\cdot k^2$, because each omitted edge $\{u, v\}$ reduces either $|d(u) - d|$ or $|d(v) - d|$ (while possibly increasing the other by one unit).

In the second stage, we insert an edge between every pair of vertices that are currently non-adjacent and have both degree smaller than d. Thus, we obtain a graph G'' that is $\frac{\epsilon' k^2/2}{k^2/2}$-close to G' such that $\{v : d_{G''}(v) < d\}$ is a clique (and $d_{G''}(v) \leq d$ for all v).

In the third stage, we iteratively increase the degrees of vertices that have degree less than d while preserving the degrees of all other vertices. Denoting by $\Gamma(v)$ the current set of neighbours of vertex v, we distinguish two cases.

Case 1: There exists a single vertex of degree less than d**.** Denoting this vertex by v, we note that $|\Gamma(v)| \leq d-2$ must hold (since $\sum_{u\in[k]} |\Gamma(u)|$ must be even, whereas in this case this sum equal $(k-1)\cdot d+|\Gamma(v)| = kd-(d-|\Gamma(v)|)$, and by the hypothesis kd is even). We shall show that there exist two vertices u and w such that $\{u, w\}$ is an edge in the current graph but $u, w \notin \Gamma(v) \cup \{v\}$. Adding the edges $\{u, v\}$ and $\{w, v\}$ to the graph, while omitting the edge $\{u, w\}$, we increase $|\Gamma(v)|$ by two, while preserving the degrees of all other vertices.

[11] Let L and H be the corresponding sets; that is, let L (resp., H) be a set of $0.01\epsilon k$ vertices having the lowest (resp., highest) degree in G. For $\ell \stackrel{\rm def}{=} \max_{v\in L}\{d_G(v)\}$ and $h \stackrel{\rm def}{=} \min_{v\in H}\{d_G(v)\}$, if $h-\ell \leq 0.04\epsilon k$, then each vertex in $[k]\setminus(L\cup H)$ has degree that resides in $\{\ell, ..., h\}$, and the claim follows (since these degrees are all within $\pm(0.02\epsilon k+1)$ from $\lfloor(\ell+h)/2\rfloor$). On the other hand, if $h-\ell > 0.04\epsilon k$, then the tester rejects with high probability (by having seen at least one vertex in L and one vertex in H, and having estimated their degrees well enough).

[12] Being even more nitpicking, we note that using $d-1$ instead of d yields an additional loss of k edges, which is OK provided $k \leq 0.01\epsilon k^2$. On the other hand, if $\epsilon < 100/k$, then we can just retrieve the entire graph using $\binom{k}{2} = O(1/\epsilon^2)$ queries.

We show the existence of two such vertices by recalling that $|\Gamma(v) \cup \{v\}| \leq d-1$ whereas all other $k-1 \geq d$ vertices in the graph have degree d (which actually implies that $k-1 \geq d+1$). Considering an arbitrary vertex $u \notin \Gamma(v) \cup \{v\}$, we note that u has d neighbors (since $u \neq v$), and these neighbors cannot all be in $\Gamma(v) \cup \{v\}$ (which has size at most $d-1$). Thus, there exists $w \in \Gamma(u) \setminus (\Gamma(v) \cup \{v\})$, and we are done.

***Case 2: There exist at least two vertices of degree less than** d.* Let v_1 and v_2 be two vertices such that $|\Gamma(v_i)| \leq d-1$ holds for both $i \in \{1,2\}$. Note that $\{v_1, v_2\}$ is an edge in the current graph, since the set of vertices of degree less than d constitute a clique. We shall show that there exists two vertices u_1 and u_2 such that $\{u_1, u_2\}$ is an edge in the current graph but neither $\{v_1, u_1\}$ nor $\{v_2, u_2\}$ are edges (and so $|\Gamma(u_1)| = |\Gamma(u_2)| = d$). Adding the edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$ to the graph, while omitting the edge $\{u_1, u_2\}$, we increase $|\Gamma(v_i)|$ by one (for each $i \in \{1,2\}$), while preserving the degrees of all other vertices.

We show the existence of two such vertices by starting with an arbitrary vertex $u_1 \notin (\Gamma(v_1) \cup \{v_1, v_2\})$. Such a vertex exists since $v_2 \in \Gamma(v_1)$ and so $|\Gamma(v_1) \cup \{v_1, v_2\}| = |\Gamma(v_1) \cup \{v_1\}| \leq d < k$. We now make the following two observations.

- Vertex u_1 has d neighbors (see above).[13] Obviously, $v_1 \notin \Gamma(u_1)$ (since $u_1 \notin \Gamma(v_1)$).
- The set $(\Gamma(v_2) \cup \{v_2\}) \setminus \{v_1\}$ has size at most $d-1$, since $v_1 \in \Gamma(v_2)$ and $|\Gamma(v_2)| < d$.

It follows that $\Gamma(u_1)$ cannot be contained in $\Gamma(v_2) \cup \{v_2\}$, since $|\Gamma(u_1) \setminus \{v_1\}| = d$ whereas $|(\Gamma(v_2) \cup \{v_2\}) \setminus \{v_1\}| \leq d-1$. Hence, there exists $u_2 \in \Gamma(u_1) \setminus (\Gamma(v_2) \cup \{v_2\})$.

Thus, in each step of the third stage, we decrease $\sum_{v \in [N]} |d_{G''}(v) - d|$ by *two units*, while preserving both the invariances established in the second stage (i.e., $\{v : d_{G''}(v) < d\}$ is a clique and $d_{G''}(v) \leq d$ for all v). Since in each step we modified three edges (and there are at most $\epsilon' k^2/2$ steps), we conclude that G'' is $\frac{3\epsilon' k^2/2}{k^2/2}$-close to a d-regular graph, and the claim follows (by recalling that G is $3\epsilon'$-close to G''). ■

Reducing the query complexity. The wasteful aspect in the aforementioned tester is that it samples $O(1/\epsilon)$ vertices and estimates the degree of each of these vertices up to an additive term of $0.01\epsilon k$. This tester admits a straightforward analysis by which if $\sum_{v \in [k]} |d_G(v) - d| > 0.05\epsilon k^2$, then at least $0.02\epsilon k$ of the vertices have degree outside the interval $[d \pm 0.03\epsilon k]$. In this analysis a vertex was defined as "exceptional" if its degree deviates from the average value by more than $0.03\epsilon k$, but when lower-bounding the number of exceptional vertices we used k as an upper bound on the contribution of each exceptional vertex (to the sum of deviations). That is, the threshold for being considered "exceptional" is minimalistic (i.e., it considers an extremely mild deviation as exceptional), but when analyzing the number of exceptional vertices we considered the maximal possible deviation.

Obviously, we must take into account both these extreme cases (i.e., both mild deviations and huge deviations of individuial degrees), but we may observe that in each case *the number of vertices of a given deviation may be related to the magnitude of the deviation.* That is, if exceptional vertices "deviate by much" (i.e., their degrees deviates from the average by at least $\delta k \gg \epsilon k$), then less samples suffice for detecting their deviation (i.e., $O(1/\delta^2) \ll O(1/\epsilon^2)$ samples suffice). On the other hand, if exceptional vertices only "deviate by little" (i.e., their degrees deviates from the average

[13] This is beacuse since $u_1 \notin \Gamma(v_1)$, whereas all vertices of degree lower than d are neighbors of v_1 (since the vertices of lower degree form a clique).

by at most $\delta k = \Omega(\epsilon k)$ (or so)), then it suffices to sample less vertices (i.e., it suffices to sample $O(\epsilon/\delta)$ vertices). Of course, we do not know which case holds, and in fact we may have a mix of several cases. Still, we can handle all cases concurrently.

Specifically, one can show that there exists $i \in [\log_2(O(1)/\epsilon)]$ such that at least $\Omega(2^{-i} \cdot k)$ of the vertices have degrees that deviate from the average by $\Theta(2^i\epsilon \cdot k/\log(1/\epsilon))$ units, since otherwise the total deviation would have been

$$\sum_{i\in[\ell]} o(2^{-i} \cdot k) \cdot \Theta(2^i\epsilon \cdot k/\log(1/\epsilon)) = \sum_{i\in[\ell]} o(\epsilon k^2/\log(1/\epsilon)) = o(\epsilon k^2)$$

in contradiction to the hypothesis. Hence, for every $i \in [\log_2(O(1)/\epsilon)]$, we attempt to detect a $\Omega(2^{-i})$ fraction of the vertices that have degrees that deviate from the average by approximately $\Theta(2^i\epsilon \cdot k/\log(1/\epsilon))$ units, where the total amount of work involved in performing the relevant estimates is

$$\sum_{i\in[\ell]} O(2^{-i})^{-1} \cdot \Theta(2^i\epsilon/\log(1/\epsilon))^{-2} = \sum_{i\in[\ell]} O(2^{-i}(\log(1/\epsilon))^2/\epsilon^2) = \widetilde{O}(1/\epsilon^2).$$

Actually, we shall obtain a slightly better result by attempting to detect a $\Omega(2^{-i})$ fraction of the vertices that have degrees that deviate from the average by approximately $\Theta(2^{4i/5}\epsilon \cdot k)$ units. (The analysis of this choice will appear within and after the presentation of Algorithm 5.2.) In addition, we simplify the analysis by introducing an auxiliary step in which we estimate the average degree of the vertices in the graph.

Algorithm 5.2 (the actual tester): *For a sufficiently large constant c, let $\ell \stackrel{\rm def}{=} \log_2(c/\epsilon)$.*

1. *The tester estimates the average degree of the graph by making $O(1/\epsilon^2)$ uniformly distributed queries. This allows to estimate the avearge degree up to $\pm\epsilon \cdot k/c$, with probability at least $5/6$. Let $\widetilde{d}$ denote the estimated average.*

2. *For every $i \in [\ell]$, the tester attempts to find a vertex with degree outside the interval $[\widetilde{d} \pm 2^{1+(4i/5)}\epsilon \cdot k/c]$, by taking a sample of $c \cdot 2^i$ vertices, and estimating their degree up to up to $\pm 2^{4i/5}\epsilon \cdot k/c$. Specifically:*

   (a) *The tester selects uniformly $c \cdot 2^i$ vertices, and estimates the degree of each of these vertices up to $\pm 2^{4i/5}\epsilon \cdot k/c$ units by using a sample of $s_i \stackrel{\rm def}{=} c^3 \cdot 2^{-3i/2}\epsilon^{-2} \gg (2^{4i/5}\epsilon/c)^{-2}$ random vertices. Note that with probability at least*

$$\begin{aligned} 1 - c \cdot 2^i \cdot \exp(-2 \cdot s_i \cdot (2^{4i/5}\epsilon/c)^2) &= 1 - c \cdot 2^i \cdot \exp(-2 \cdot c^3 2^{-3i/2}\epsilon^{-2} \cdot 2^{8i/5}\epsilon^2/c^2) \\ &= 1 - c \cdot 2^i \cdot \exp(-2c \cdot 2^{i/10}) \\ &> 1 - 2^{-i-c} \end{aligned}$$

   all these estimates are as desired.

   (b) *If any of these estimates is outside the interval $[\widetilde{d} \pm 2^{1+(4i/5)}\epsilon \cdot k/c]$, then the tester rejects.*

If the tester did not reject in any of these ℓ iterations, then it accepts.

The query complexity of Algorithm 5.2 is $O(1/\epsilon^2)+\sum_{i\in[\ell]} c2^i \cdot c^3 2^{-3i/2}\epsilon^{-2} = O(1/\epsilon^2)$. The probability that any of the estimates performed in (any of the iterations of) Step 2 deviates by more than desrired is $\sum_{i\in[\ell]} 2^{-i-c} = 2^{-c} < 1/10$.

We first observe that Algorithm 5.2 accepts each regular graph with probability at least 2/3. This is the case since, $\mathbf{Pr}[|\widetilde{d}-d| \leq \epsilon k/c] \geq 0.9$, where d denotes the degree of each vertex in the graph, and with probability at least 0.9 for each $i \in [\ell]$ each of the degree estimates performed in (the $i^{\rm th}$ iteration of) Step 2 fell inside the interval $[d \;\pm\; 2^{4i/5}\epsilon \cdot k/c]$, which is contained in $[\widetilde{d} \;\pm\; 2^{1+(4i/5)}\epsilon \cdot k/c]$.

On the other hand, if a graph G is accepted with probability at least 1/3, then (as detailed next), for every $i \in [\ell]$, it holds that all but at most a 2^{-i} fraction of the vertices have degree that is within $2^{2+(4i/5)}\epsilon \cdot k/c$ of the average degree of G, denoted d.

> Claim: If, for some $i \in [\ell]$, more than a 2^{-i} fraction of the vertices have degree that deviates from d by more than $2^{2+(4i/5)}\epsilon \cdot k/c$, then Algorithm 5.2 rejects with probability greater than 2/3.
>
> Proof: We first observe that, with probability at least 0.9, such a deviating vertex, denoted v, is selected in the $i^{\rm th}$ iteration of Step 2. Now, with probability at least 0.9, the degree v is estimated within $\pm 2^{4i/5}\epsilon \cdot k/c$ of its correct value. Recalling that $\mathbf{Pr}[|\widetilde{d}-d| < \epsilon k/c] \geq 0.9$, we conclude that, with probability at least 0.7, the estimated degree of v deviates from $\widetilde{d}$ by more than $2^{2+(4i/5)}\epsilon k/c - 2^{4i/5}\epsilon k/c - \epsilon k/c \geq 2^{1+(4i/5)}\epsilon k/c$, which causes the algorithm to reject, and the claim follows. ■

Now, for each $i \in [\ell]$, let us denote the set of deviating vertices by B_i; that is, each vertex in $[k]\setminus B_i$ has degree in $(d \pm 2^{2+(4i/5)}\epsilon/c \cdot k)$. Recall that $|B_i| \leq 2^{-i} \cdot k$. (Also, let $B_0 = [k]$, and note that $[k] \setminus B_\ell = \cup_{i\in[\ell]}(B_{i-1} \setminus B_i)$.)[14] Hence,

$$\begin{aligned}
\sum_{v\in[k]\setminus B_\ell} |d_G(v)-d| &= \sum_{i\in[\ell]} \sum_{v\in B_{i-1}\setminus B_i} |d_G(v)-d| \\
&\leq \sum_{i\in[\ell]} |B_{i-1}| \cdot \max_{v\in[k]\setminus B_i}\{|d_G(v)-d|\} \\
&\leq \sum_{i\in[\ell]} 2^{-(i-1)} \cdot 2^{2+(4i/5)}\epsilon k^2/c \\
&= \sum_{i\in[\ell]} 2^{-0.2i} \cdot 8\epsilon k^2/c
\end{aligned}$$

which is smaller than $0.04\epsilon k^2$ by a suitable choice of c. Finally, under such a choice, $|B_\ell| \leq 2^{-\ell} \cdot k = (\epsilon/c) \cdot k$ is smaller than $0.01\epsilon k$, hence $\sum_{v\in B_\ell} |d_G(v)-d| < 0.01\epsilon k^2$, and so $\sum_{v\in[k]} |d_G(v)-d| < 0.05\epsilon k^2$. Applying Claim 5.1, it follows that G is 0.3ϵ-close to being regular, and the theorem follows. ■

[14] Indeed, the definition of B_0 is a fictitious; it is made in order to have $[k] \setminus B_\ell = \cup_{i\in[\ell]}(B_{i-1} \setminus B_i)$ hold. The alternative would have been to treat the case of $i=1$ separately; that is, write $[k]\setminus B_\ell = ([k]\setminus B_1) \cup \cup_{i=2}^{\ell}(B_{i-1}\setminus B_i)$. Note that, either way, we treat B_ℓ separately.

2.4 Digest: Levin's economical work investment strategy

The strategy underlying Algorithm 5.2 can be traced to Levin's work on one-way functions and pseudorandom generators [40]. An attempt to abstract this strategy follows.

The strategy refers to situations in which one can sample a huge space that contains elements of different quality such that elements of lower quality require more work to utilize. The aim is to utilize some element, but the work required for utilizing the various elements is not known a priori, and it only becomes known after the entire amount of required work is invested. Only a lower bound on the expected quality of elements is known, and it is also known how the amount of required work relates to the quality of the element (see specific cases below). Note that it may be that most of the elements are of very poor quality, and so it is not a good idea to select a single (random) element and invest as much work as is needed to utilize it. Instead, one may want to select many random elements and invest in each of them a limited amount of work (which may be viewed as probing the required amount of work).

To be more concrete, let us denote the (unknown to us) quality of a sample point $\omega \in \Omega$ by $q(\omega) \in (0,1]$, and suppose that the amount of work that needs to be invested in a sample point ω is $O(1/q(\omega)^c)$, where in the setting of Algorithm 5.2 it holds that $c = 2$. Indeed, $c = 1$ and $c = 2$ are the common cases, where $O(1/q(\omega))$ corresponds to the number of trials that is required to succeed in an experiment (which depends on ω) that succeeds with probability $q(\omega)$, and $O(1/q(\omega)^2)$ corresponds to the number of trials that is required for estimating the success probability of an experiment up to $\pm q(\omega)$. Recall that we only know a lower bound, denoted ϵ, on the average quality of an element (i.e., $\mathbb{E}_{\omega\in\Omega}[q(\omega)] > \epsilon$), and we wish to minimize the total amount of work invested in utilizing some element.

One natural strategy that comes to mind is to sample $O(1/\epsilon)$ points and invest $O(1/\epsilon^c)$ work in each of these points. In this case we succeed with constant probability, while investing $O(1/\epsilon^{c+1})$ work. The analysis is based on the fact that $\mathbb{E}_\omega[q(\omega)] > \epsilon$ implies that $\mathbf{Pr}_\omega[q(\omega) > \epsilon/2] > \epsilon/2$. The strategy underlying Algorithm 5.2 is based on the fact that there exists $i \in [\log_2(O(1)/\epsilon)]$, such that $\mathbf{Pr}_\omega[q(\omega) > 2^{4i/5} \cdot \epsilon] = \Omega(2^{-i})$. In this case (when $c = 2$), for every i, we selected $O(2^i)$ points and invested $O(1/2^{4i/5}\epsilon)^2$ work in each of them. Hence, we achieved the goal while investing $(1/\epsilon^2)$ work.

Teaching note: In the following general analysis, we shall use a setting of parameters that is different from the one used above. This is made in order to better serve the case of $c = 1$. In addition, we believe that a different variation on the very same idea will serve the reader better.

In general, for any $c \geq 1$ and $\ell = \lceil \log_2(2/\epsilon) \rceil$, we may use the fact that there exists $i \in [\ell]$ such that $\mathbf{Pr}_\omega[q(\omega) > 2^i \cdot \epsilon] > 2^{-i}/(i+3)^2$. (The analysis is analogous to the one performed at the end of the proof of Theorem 5, although the quantity analyzed here is different (and so are some parameters).)[15] Hence, selecting $O(i^2 \cdot 2^i)$ points (for each $i \in [\ell]$), and investing $O(1/2^i\epsilon)^c$ work

[15] Let $B_i = \{\omega \in \Omega : q(\omega) > 2^i\epsilon\}$ and $B_0 = \Omega$, and note that $B_\ell = \emptyset$. Suppose, towards the contradiction, that $|B_i| \leq 2^{-i}/(i+3)^2$ for every $i \in [\ell]$. Then,

$$\begin{aligned}\sum_{\omega\in\Omega} q(\omega) &= \sum_{i\in[\ell]} \sum_{\omega\in B_{i-1}\setminus B_i} q(\omega) \\ &\leq \sum_{i\in[\ell]} |B_{i-1}| \cdot 2^i\epsilon\end{aligned}$$

in each of them, we achieved the goal while investing a total amount of work that equals

$$\sum_{i\in[\ell]} O(i^2 \cdot 2^i/(2^i\epsilon)^c) = O(1/\epsilon^c) \cdot \sum_{i\in[\ell]} i^2 \cdot 2^{-(c-1)\cdot i}$$

which equals $(1/\epsilon^c)$ work if $c > 1$ and $\widetilde{O}(1/\epsilon)$ work if $c = 1$. (For $c > 1$ we use $\sum_{i\in[\ell]} 2^{-\Omega(i)} = O(1)$, whereas for $c = 1$ we use $\sum_{i\in[\ell]} i^2 = O(\ell^3)$.) The same argument extends to the case that the work invested in ω is $\widetilde{O}(1/q(\omega)^c)$; see Exercise 2.

3 Graph Partition Problems

In this section we present a natural class of graph properties, called general graph partition problems, which contains properties such as k-`Colorability` (for any $k \geq 2$) and properties that refer to the density of the max-clique and to the density of the max-cut in a graph. The main result of this section is that each of these properties has a tester of query complexity that is polynomial in the reciprocal of the proximity parameter.

Loosely speaking, a graph partition problem calls for partitioning the graph into a given number of parts such that the sizes of the parts fit the given bounds and ditto with respect to the number of edges between parts. More specifically, each graph partition problem (resp., property) is specified by a number $t \in \mathbb{N}$ and a sequence of intervals (which serve as parameters of the problem), and a graph $G = ([k], E)$ is a YES-instance of this problem (resp., has the corresponding property) if there exists a t-partition, $(V_1, ..., V_t)$, of $[k]$ such that

1. For each $i \in [t]$, the density of V_i fits the corresponding interval (specified in the sequence of parameters).
2. For each $i, j \in [t]$ (including the case $i = j$), the density of edges between V_i and V_j fits the corresponding interval.

A formal definition of this framework is deferred to Section 3.2; here we only clarify the framework by considering a few appealing examples that refer to the case of $t \leq 2$.

We start by considering the case of $t = 1$, which is a bit of "abuse" of the term partition. Two natural properties that can be casted in that case are the property of being a clique and the property of having at least $\rho \cdot k^2$ edges, for any $\rho \in (0, 0.5)$. The first property can be ϵ-tested by uniformly selecting $O(1/\epsilon)$ vertex-pairs and checking if each of these pairs constitutes an edge of the graph. The second property can be ϵ-tested by estimating the fraction of edges in the graph, up to an additive deviation of $\epsilon/2$, which can be done using a random sample of $O(1/\epsilon^2)$ vertex-pairs. Turning to the case of $t = 2$, we consider the following natural properties.

Biclique: A graph $G = ([k], E)$ is a biclique (a.k.a a complete bipartite graph) if its vertices can be 2-partitioned into two parts, denoted V_1 and V_2, such that each part is an independent set and all pairs in $V_1 \times V_2$ are connected in the graph (i.e., $E = \{\{u, v\} : (u, v) \in V_1 \times V_2\}$).

$$\begin{aligned} &\leq \sum_{i\in[\ell]} 2^{-(i-1)}|\Omega| \cdot 2^i\epsilon/((i-1)+3)^2 \\ &< 2\epsilon|\Omega|/2 \end{aligned}$$

where the last inequality uses $\sum_{i\geq 1} \frac{1}{(i+t)^2} < \sum_{i\geq 1} \frac{1}{(i+t)(i+t-1)}$, which equals $\sum_{i\geq 1} \left(\frac{1}{i+t-1} - \frac{1}{i+t}\right) = 1/t$.

Bipartiteness: A graph $G = ([k], E)$ is bipartite (or 2-colorable) if its vertices can be 2-partitioned into two parts, V_1 and V_2, such that each part is an independent set (i.e., $E \subseteq \{\{u, v\} : (u, v) \in V_1 \times V_2\}$).

Max-Cut: For $\rho \in (0, 0.25]$, a graph $G = ([k], E)$ has a ρ-cut if its vertices can be 2-partitioned into two parts, V_1 and V_2, such that the number of edges between V_1 and V_2 is at least $\rho \cdot k^2$ (i.e., $|E \cap \{\{u, v\} : (u, v) \in V_1 \times V_2\}| \geq \rho \cdot k^2$).

Min-Bisection: For $\rho \in (0, 0.25]$, a graph $G = ([k], E)$ has a ρ-bisection if its vertices can be 2-partitioned into two equal sized parts, V_1 and V_2, such that the number of edges between V_1 and V_2 is at most $\rho \cdot k^2$ (i.e., $|V_1| = |V_2|$ and $|E \cap \{\{u, v\} : (u, v) \in V_1 \times V_2\}| \leq \rho \cdot k^2$).

Max-Clique: For $\rho \in (0, 1]$, a graph $G = ([k], E)$ has a ρ-clique if its vertices can be 2-partitioned into two parts, V_1 and V_2, such that $|V_1| = \lceil \rho \cdot k \rceil$ and the subgraph induced by V_1 is a clique (i.e., for every distinct $u, v \in V_1$ it holds that $\{u, v\} \in E$).

Indeed, with the exception of Max-Clique, all the foregoing properties generalized naturally to the case of $t > 2$. As stated in the beginning of this section, all of these properties are ϵ-testable using $\text{poly}(1/\epsilon)$ queries (for details see Section 3.1 and 3.2). Here we consider the case of `Biclique`.

Proposition 6 (testing whether a graph is a biclique (in the dense graph model)): *The property* `Biclique` *has a* (one-sided error) *proximity oblivious tester that makes three queries and has linear rejection probability. That is, a graph that is ϵ-far from being a biclique is rejected with probability at least $\Omega(\epsilon)$, whereas a biclique is accepted with probability 1.*

We stress that the empty graph $G = ([k], \emptyset)$ is considered a biclique (by virtue of a trivial 2-partition $([k], \emptyset)$). Note that ϵ-testing that a graph is not empty can be done by $O(1/\epsilon)$ queries (see Proposition 3).

Proof: The tester selects uniformly three random vertices and accepts if and only if the induced subgraph is a biclique (i.e., contains either two edges or no edges).[16] We stress that while the selected vertices are uniformly and independently distributed in $[k]$, the queried pairs are dependent (although each query is uniformly distributed in $[k] \times [k]$).

If $G = ([k], E)$ is a biclique, then it is accepted with probability 1, since the induced graph is a 3-vertex biclique. In other words, if all three vertices were selected in the same independent set of the k-vertex biclique, then the induced subgraph is a 3-vertex independent set (which is a biclique), and otherwise (i.e., when one selected vertex resides in one independent set and the other two vertices reside in the other set) the induced subgraph is a 3-vertex biclique with two edges.

Assuming that G is ϵ-far from being a biclique, fix the first vertex u that is selected by the tester. Then, u defines a 2-partition of the vertices of G such that the neighbours of u are on one side and the other vertices are on the other; that is, the 2-partition is $(\Gamma(u), [k] \setminus \Gamma(u))$, where $\Gamma(u) = \{v \in [k] : \{u, v\} \in E\} \not\ni u$. Since G is ϵ-far from being a biclique, there are at least ϵk^2 vertex pairs[17] that *violate* this 2-partition, where a pair (v, w) is said to violate the 2-partition

[16] This description ignores the possibility that the selected vertices are not distinct. In such a case, we just accept without making any queries. Alternatively, we can select uniformly a 3-subset of $[k]$.

[17] Note that here we count ordered pairs of vertices, rather than unordered pairs. Indeed, at some times it is more convenient to count in one way, and at other times the other way is preferred. We believe that, when low level details are concerned, local convenience should have precedence over global consistency.

$(\Gamma(u), [k] \setminus \Gamma(u))$ if the subgraph induced by $\{u, v, w\}$ is not a biclique. (That is, a violating pair represents either an edge that is missing between the two parts (i.e., between $\Gamma(u)$ and $[k] \setminus \Gamma(u)$) or an edge that is present inside one of these parts (i.e., internal to either $\Gamma(u)$ or $[k] \setminus \Gamma(u)$).) Hence, the probability that the tester selects a violating pair is at least $\frac{\epsilon k^2}{k^2}$, and the claim follows. ■

Digest. The analysis of the foregoing tester reveals that we can actually select the first vertex arbitrarily, and only select the two other vertices at random. More importantly, the foregoing proof illustrated a technique that is quite popular in the area (see, e.g., Section 3.1). Specifically, the first vertex "induces" (or forces) auxiliary conditions on the graph (i.e., the existence of edges between its neighbors and non-neighbors and the non-existence of other edges), and these conditions are checked by the random pair of vertices selected next. In general, in the "force and check" technique, the tester designates parts of its sample to force conditions on the object, and these conditions are checked by the second part of the sample. Note that the forcing can be implicit (like the partition of $[k]$ according to neighbors versus non-neighbors of u), whereas the checking actually tests these conditions via queries (e.g., the three queries of the foregoing tester are defined and performed only once the other two vertices are selected).

Teaching note: The following four paragraphs may be used as a motivation towards the tester for `Bipartiteness` (of Section 3.1), but some readers may find this discussion a bit too abstract.

Turning back to the tester presented in the proof of Proposition 6, recall that the vertex u induced a 2-partition of $[k]$ and that the placement of each vertex v with respect to that partition can be determined by a single query to G. In other words, we have implemented an oracle $\chi : [k] \to \{1, 2\}$ (i.e., $\chi(v) = 1$ if and only if $v \in \Gamma(u)$ (or equivalently, if and only if $\{v, u\} \in E$)), and observed that G is a biclique if and only if χ is a 2-partition that witnesses this claim (i.e., $E = \{\{v, w\} : \chi(v) \neq \chi(w)\}$). We then checking if G is a biclique by selecting a random pair (v, w) and accepted if and only if $\{u, v\} \in E \iff \chi(v) \neq \chi(w)$.

As a motivation towards the presentation of the tester for `Bipartiteness`, suppose that one provides an implementation of T oracles $\chi_1, ..., \chi_T : [k] \to \{1, 2\}$ and shows that G *is a bipartite if and only if at least one of these* χ_i*'s is a 2-partition that witnesses this claim* (i.e., $E \subseteq \{\{v, w\} : \chi_i(v) \neq \chi_i(w)\}$). Then, we can test whether G is bipartite or ϵ-far from being bipartite by selecting $m = O(\epsilon^{-1} \log T)$ random pairs $(v_1, w_1), ..., (v_m, w_m)$ and accepting if and only if there exists an $i \in [T]$ such that for every $j \in [m]$ it holds that $\{u, v\} \in E \implies \chi_i(v_j) \neq \chi_i(w_j)$.[18] Furthermore, if we can answer all these Tm queries by making a total number of $q(\epsilon)$ queries to the graph G, then we would get an ϵ-tester of query complexity $q(\epsilon)$. As shown next, this would follow even if we can only answer these oracle queries for vertices in a (good) set V, provided that all but at most $0.1\epsilon k^2$ of the edges are adjacent to vertices in V (where and edge is considered adjacent to V if both its endpoint are adjacent to some vertices in V).

The tester operates as outlined above, except that whenever it gets no answer to $\chi_i(v)$ (i.e., $v \notin V$), it just sets $\chi_i(v)$ so to avoid rejection (whenever possible). This provision guarantees that the tester always accepts a bipartite graph (since for the suitable χ_i there exists a setting of $\chi_i(v)$ (for every $v \in [k] \setminus V$) that avoids rejection). On the other hand, if G is ϵ-far from being bipartite, then for every $\chi : [k] \to \{1, 2\}$ there exist at least ϵk^2 pairs (v, w) such that $\{v, w\} \in E$

[18] See analysis in the end of last paragraph.

and $\chi(v) = \chi(w)$. In particular, this holds for each of the foregoing χ_i's, whereas only $0.2\epsilon k^2$ of these pairs may be "invisible" to the tester.[19] Hence, each χ_i is detected as bad with probability at least $1 - (1 - 0.8\epsilon)^m = 1 - (1/3T)$.

The crucial details that were avoided so far are the specification of the T partitions χ_i's and their implementation via queries to the graph. We leave these crucial details to the proof of Lemma 8, since it makes little sense to give these details without provide that they actually work.[20]

3.1 Testing Bipartiteness

We first note that `Bipartiteness` has no proximity oblivious tester that makes a constant number of queries (and has rejection probability that only depends on the distance of the graph from being bipartite).[21] This can be shown by considering graphs that have "odd-girth" that is larger than the potential query complexity (see Exercise 4). Nevertheless, testing `Bipartitenss` is quite simple: It amounts to selecting a small random set of vertices, and checking whether the induced subgraph is bipartite. Specifically, the size of the sample is polynomial in the reciprocal of the proximity parameter.

Algorithm 7 (testing `Bipartiteness` in the dense graph model): *On input k, ϵ and oracle access to an adjacency predicate of an k-vertex graph, $G = ([k], E)$, the tester proceeds as follows:*

1. *Uniformly select a subset of $\widetilde{O}(1/\epsilon^2)$ vertices of G.*

2. *Accept if and only if the subgraph induced by this subset is bipartite.*

Step (2) amounts to querying the adjacency predicate on all pairs of vertices that belong to the subset selected at Step (1), and testing whether the induced subgraph is bipartite (e.g., by running BFS).[22] As will become clear from the analysis, it actually suffice to query only $\widetilde{O}(1/\epsilon^3)$ of these pairs. Since being bipartite is "closed under taking subgraph" (i.e., if G is bipartite then every subgraph of G is bipartite), Algorithm 7 always accepts bipartite graphs (i.e., it has one-sided error as a tester). Hence, in case of rejection, the algorithm can output a witness of length $\mathrm{poly}(1/\epsilon)\log k$ that certifies that the graph is not bipartite.[23] The analysis of Algorithm 7 is completed by the following lemma.

Lemma 8 (analysis of Algorithm 7): *If $G = ([k], E)$ is ϵ-far from being bipartite, then Algorithm 7 rejects it with probability at least $1/2$, when invoked with the proximity parameter ϵ.*

[19] Recall that the number of edges that have at least one endpoint that does not neighbor V is at most $0.1\epsilon k^2$.

[20] If one insists to know, then the answer is essentially as follows. For a random set U of size $t = \widetilde{O}(1/\epsilon)$, we consider all 2-partitions of U, and, for such each 2-partition (U_1, U_2), we define the 2-partition $\chi_{U_1,U_2} : [k] \to \{1, 2\}$ such that $\chi_{U_1,U_2}(v) = i$ if any only if v is a neighbor of some vertex in U_{3-i}. Note that this definition may be contradictory (when v neighbors both U_1 and U_2) and partial (if v neighbors no vertex in U). Both issues will be handled in the proof of Lemma 8.

[21] Recall that the definition of proximity oblivious tester that we used in this text requires that the rejection probability only depends on the distance of the input from the property.

[22] Recall that a connected graph is bipartite if and only if for any vertex v there is no edge between any pair of vertices that are at equal distance from v. (Indeed, the existence of such edge implies the existence of an odd cycle, and otherwise we can place all vertices that are at odd distance from v in the same side of the 2-partition.)

[23] Indeed, in this case, the witness may consist of an odd-length cycle of $\widetilde{O}(1/\epsilon^2)$ vertices.

Proof: Denoting by R the random $\widetilde{O}(1/\epsilon^2)$-subset of $[k]$ selected in Step (1), we shall show that, with probability at least 1/2, the subgraph of G induced by R is not bipartite. That is, assuming that G is ϵ-far from bipartite, we prove that with high probability G_R is not bipartite, where G_R is the subgraph of G induced by R.

We view R as a union of two disjoint sets U and S, where $t \stackrel{\rm def}{=} |U| = O(\epsilon^{-1} \cdot \log(1/\epsilon))$ and $m \stackrel{\rm def}{=} |S| = O(t/\epsilon)$. We will consider all possible 2-partitions of U, and associate a partial 2-partition of $[k]$ with each such 2-partition of U. Specifically, the partial 2-partition of $[k]$ that is associated with a given 2-partition (of U), denoted (U_1, U_2), places all neighbors of U_1 (respectively, U_2) opposite to U_1 (respectively, U_2).[24] The point is that such a placement of vertices is forced upon any 2-partition that is consistent with the 2-partition (U_1, U_2) in the sense that if v neighbors U_i and the subgraph induced by $U \cup \{v\}$ is bipartite with a 2-partition that places U_1 on one side and U_2 on the other, then v must be on the side opposite to U_i.

The idea is that since G is ϵ-far from being bipartite, then any 2-partition of its vertices (and, in particular, one associated to the 2-partition of U) must have at least $\epsilon k^2/2$ edges that are internal to one of the sides of the said 2-partition of $[k]$, and (with high probability) the sample S will hit some of these edges. There are a couple of problems with this idea. Firstly, we do not know the 2-partition of U, but as hinted above we shall consider all of them. (Indeed, there are only 2^t possibilities, whereas the size of S is selected such that the probability of not detecting a problem with any fixed 2-partition is smaller than $2^{-t}/10$.) Secondly, the 2-partition of U only forces the placement of vertices that neighbour U, while we do not know the placement of the other vertices (and so cannot detect problems with edges incident to them).

The second problem is solved by showing that, with high probability, almost all high-degree vertices in $[k]$ do neighbor U, and so are forced by each of its possible 2-partitions. Since there are relatively few edges incident to vertices that do not neighbor U, it follows that, with very high probability, each such 2-partition of U is detected as illegal by G_R. Details follow, but before we proceed let us stress the key observation: *It suffices to rule out relatively few* (partial) *2-partitions of* $[k]$ (i.e., those induced by 2-partitions of U), rather than all possible 2-partitions of $[k]$.

We use the notations $\Gamma(v) \stackrel{\rm def}{=} \{u : \{u,v\} \in E\}$ and $\Gamma(X) \stackrel{\rm def}{=} \cup_{v \in X} \Gamma(v)$. Given a 2-partition (U_1, U_2) of U, we define a (possibly partial) 2-partition of $[k]$, denoted (V_1, V_2), such that $V_1 \stackrel{\rm def}{=} \Gamma(U_2)$ and $V_2 \stackrel{\rm def}{=} \Gamma(U_1)$, where we assume, for simplicity that $V_1 \cap V_2$ is indeed empty (otherwise things are easier).[25] As suggested above, if one claims that G can be "legally bi-partitioned" with U_1 and U_2 on different sides, then $V_1 = \Gamma(U_2)$ must be on the opposite side to U_2 (and $\Gamma(U_1)$ opposite to U_1).[26] Note that the 2-partition of U places no restriction on vertices that have no neighbor in U. Thus, we first ensure that *almost all* "influential" (i.e., "high-degree") vertices in $[k]$ have a neighbor in U.

Definition 8.1 (high-degree vertices and good sets): *We say that a vertex v is of* high-degree *if it has degree at least $\epsilon k/6$. We call U* good *if all but at most $\epsilon k/6$ of the high-degree vertices have a neighbor in U.*

[24] Indeed, the placement of vertices that do not neighbor U remains undetermined (or is arbitrary). This is the reason that we referred to the associated partition as partial.

[25] In this case the 2-partition (U_1, U_2) is ruled out by G_U. In the rest of the analysis, we shall not use this fact. The reader may redefine $V_2 = \Gamma(U_1) \setminus V_1$.

[26] Formally, we say that for any 2-coloring $\chi : [k] \to \{1,2\}$ (i.e., a mapping χ such that $\chi(u) \neq \chi(v)$ for every $\{u,v\} \in E$), the following holds: if $\chi(u) = i$ for every $u \in U_i$ and $i \in \{1,2\}$, then $\chi(v) \neq i$ for every $v \in \Gamma(U_i)$.

We comment that NOT insisting that a good set U neighbors *all* high-degree vertices allows us to show that, with high probability, a random U of size $\widetilde{O}(1/\epsilon)$ is good, where the point is that this size is unrelated to the size of the graph. (In contrast, if we were to insist that a good U neighbors *all* high-degree vertices, then we would have had to use $|U| = \Omega(\epsilon^{-1} \log k)$.)

Claim 8.2 (random t-sets are good): *With probability at least* $3/4$*, a uniformly chosen set* U *of size* t *is good.*

Proof: For any high-degree vertex v, the probability that v does not have any neighbor in a uniformly chosen U is at most $(1-(\epsilon/6))^t < \epsilon/24$, since $t = \Omega(\epsilon^{-1}\log(1/\epsilon))$. Hence, the expected number of high-degree vertices that do not have a neighbor in a random set U is less than $\epsilon k/24$, and the claim follows by Markov's Inequality. ■

Definition 8.3 (disturbing a 2-partition of U): *We say that an edge* disturbs the 2-partition (U_1, U_2) of U *if both its end-points are in the same set* $\Gamma(U_i)$*, for some* $i \in \{1, 2\}$*.*

Claim 8.4 (lower bound on the number of disturbing edges): *For any good set* U *and any 2-partition of* U*, at least* $\epsilon k^2/6$ *edges disturb this 2-partition.*

Proof: Since G is ϵ-far from being bipartote, each 2-partition of $[k]$ has at least $\epsilon k^2/2$ violating edges (i.e., edges with both end-points on the same side). In particular, this holds for the 2-partition (V_1, V_2) defined by letting $V_1 = \Gamma(U_2)$ and $V_2 = [k] \setminus V_1$, where (U_1, U_2) is the given 2-partition of U. We upper bound the number of edges with both sides in the same V_i that are not disturbing. Actually, we upper bound the number of edges that have an end-point that is not in $\Gamma(U)$.

- The number of edges incident at high-degree vertices that do not neighbor the good set U is bounded by $(\epsilon k/6) \cdot k$, since there are at most $\epsilon k/6$ such vertices.
- The number of edges incident at vertices that are not of high-degree is bounded by $k \cdot \epsilon k/6$, since each such vertex has at most $\epsilon k/6$ incident edges.

Hence, that are at most $\epsilon k^2/3$ edges that do not have both end-points in $\Gamma(U)$. This leaves us with at least $\epsilon k^2/6$ violating edges with both end-points in $\Gamma(U)$ (i.e., edges disturbing the 2-partition (U_1, U_2)). ■

The lemma follows by observing that G_R is bipartite only if either (1) the set U is not good; or (2) the set U is good and there exists a partition of U so that none of the disturbing edges occurs in G_R. Using Claim 8.2 the probability of event (1) is bounded by $1/4$, whereas the probability of event (2) is bounded by the probability that there exists a 2-partition of U such that none of the corresponding disturbing edges has both end-points in the second sample S. By Claim 8.4, each 2-partition of U has at least $\epsilon k^2/6$ disturbing edges, and (as shown next) the probability that none of them has both end-points in S is at most $(1-(\epsilon/6))^{m/2}$. Actually, we pair the m vertices of S, and consider the probability that none of these $m/2$ pairs is a disturbing edge for some partition of U. Thus, the probability of event (2) is upper-bounded by

$$2^t \cdot \left(1 - \frac{\epsilon}{6}\right)^{m/2} < \frac{1}{4}$$

where the inequality holds since $m = \Omega(t/\epsilon)$. The lemma follows. ■

Approximate 2-coloring procedures that arises from the proof of Lemma 8. By an approximate 2-coloring of a graph $G = ([k], E)$, we mean a 2-partition $\chi : [k] \to \{1, 2\}$ with relatively few edges have endpoints that are assigned the same color (e.g., $|\{\{u, v\} \in E : \chi(v) = \chi(w)\}| = o(|E|)$). The partitioning rule employed in the proof of Lemma 8 (i.e., $\chi(v) = 1$ if and only if $v \in \Gamma(U_2)$ for an adequate 2-partition (U_1, U_2) of U) yields a randomized $\text{poly}(1/\epsilon) \cdot k$-time algorithm for approximately 2-coloring a k-vertex bipartite graph such that (with high probability) at most ϵk^2 edges have endpoints that are assigned the same color. This is done by running the tester, determining a 2-partition of U that is consistent with any 2-coloring of the subgraph induced by $R = U \cup S$, and 2-partitioning $[k]$ as done in the proof (with vertices that do not neighbor U, or neighbor both U_1 and U_2, placed arbitrarily). Thus, the placement (or coloring) of each vertex is determined by inspecting at most $\widetilde{O}(1/\epsilon)$ entries of the adjacency matrix. Furthermore, the aforementioned 2-partition of U constitutes a succinct representation of the 2-partition of the entire graph. All this is a typical consequence of the fact that the analysis of the tester follows the "force-and-check" paradigm.

On the complexity of testing `Bipartiteness`. We comment that a more complex analysis, due to Alon and Krivelevich [4], implies that the Algorithm 7 is an ϵ-tester for `Bipartiteness` even if one selects only $\widetilde{O}(1/\epsilon)$ vertices (rather than $\widetilde{O}(1/\epsilon^2)$ vertices) in Step (1)). That is, *if G is ϵ-far from being bipartite, then, with high probability, the subgraph induced by a random set of $\widetilde{O}(1/\epsilon)$ vertices of G is not bipartite.* The reader can verify that inspecting the subgraph induced by $o(1/\epsilon)$ vertices will not do (see Exercise 5). Furthermore, while the result of Alon and Krivelevich [4] asserts that `Bipartiteness` can be ϵ-tested using $\widetilde{O}(1/\epsilon^2)$ *non-adaptive* queries, Bogdanov and Trevisan [15] showed that $\Omega(1/\epsilon^2)$ queries are required by any *non-adaptive* ϵ-tester. For general (adaptive) testers, a lower bound of $\Omega(1/\epsilon^{3/2})$ queries is known [15], even if the input (k-vertex) graph has max-degree at most $O(\epsilon k)$, and this lower bound is almost tight for that case [35].

Open Problem 9 (what is the query complexity of testing `Bipartiteness`): *Can* `Bipartiteness` *be ϵ-tested using $\widetilde{O}(1/\epsilon^c)$ queries for some $c < 2$? And how about $c = 1.5$?*

We mention that Bogdanov and Li [14] showed that *the answer to the first question is positive, provided that the following conjecture holds.*

Conjecture 10 (a random induced subgraph preserves the distance from being bipartite): *If G is ϵ-far from being bipartite, then, with probability at least $2/3$, the subgraph induced by a set of $\widetilde{O}(1/\epsilon)$ vertices of G is $\Omega(\epsilon)$-far from being bipartite.*

Recall that Alon and Krivelevich [4] showed that, with high probability, such a subgraph is not bipartite; but the conjecture postulates that it is far from being bipartite. Note that the proof of Lemma 8 implies that (with high probability) the subgraph induced by a set of $\widetilde{O}(1/\epsilon^2)$ vertices of G is $\Omega(\epsilon)$-far from being bipartite (see Exercise 6).

3.2 The actual definition and the general result

It is time to provide the actual definition of the class of *general graph partition problems.* Recall that a graph partition problem calls for partitioning the graph into a predetermined number of parts such that the sizes of the parts fit predetermined bounds and ditto with respect to the number of edges between parts. Hence, each problem (or property) in this class is defined in terms of a sequence of parameters. The main parameter, denoted t, represents the number of parts in the partition. In addition, we have, (1) for each $i \in [t]$, a pair of corresponding upper and lower bounds on the density of the i^{th} set, and (2) for each $(i,j) \in [t]^2$, two pairs of corresponding upper and lower bounds on the "absolute" and "relative" density of the edges between the i^{th} and j^{th} sets, where by absolute (resp., relative) density we mean the size normalized by k^2 (resp., by the maximum number possible, given the actual sizes of the i^{th} and j^{th} sets).

In the following definition, for a graph $G = (V, E)$ and two sets $V', V'' \subseteq V$, we denote by $E(V', V'')$ the set of edges having one endpoint in V' and another endpoint in V''. (Indeed, if $V' = V''$, then $E(V', V'')$ denotes the set of edges with both endpoints in $V' = V''$.) Note that, for $V' \cap V'' = \emptyset$, it holds that $|E(V', V'')| \le |V'| \cdot |V''|$, whereas $|E(V', V')| \le \binom{|V'|}{2}$. For that reason (and for it only), Conditions 3 and 4 are separated.[27]

Definition 11 (general partition problem): *A* graph partition problem *is parameterized by a sequence $(t, (L_i, H_i)_{i\in[t]}, (L^{\texttt{abs}}_{i,j}, H^{\texttt{abs}}_{i,j})_{i,j\in[t]}, (L^{\texttt{rel}}_{i,j}, H^{\texttt{rel}}_{i,j})_{i,j\in[t]})$ and consists of all graphs $G = (V, E)$ such that there exists a t-partition of V, denoted $(V_1, ..., V_t)$, that satisfies the following conditions:*

1. *For every $i \in [t]$,*
$$L_i \le \frac{|V_i|}{|V|} \le H_i.$$

2. *For every $i, j \in [t]$,*
$$L^{\texttt{abs}}_{i,j} \le \frac{|E(V_i, V_j)|}{|V|^2} \le H^{\texttt{abs}}_{i,j}.$$

3. *For every $i, j \in [t]$ such that $i \ne j$,*
$$L^{\texttt{rel}}_{i,j} \le \frac{|E(V_i, V_j)|}{|V_i| \cdot |V_j|} \le H^{\texttt{rel}}_{i,j}.$$

4. *For every $i \in [t]$,*
$$L^{\texttt{rel}}_{i,i} \le \frac{|E(V_i, V_i)|}{\binom{|V_i|}{2}} \le H^{\texttt{rel}}_{i,i}.$$

Definition 11 extends the definition used in [26, Sec. 9], which only contained Conditions 1 and 2. We believe that the added conditions (Nr. 3 and 4) increase flexibility and avoid some technicalities. Using Definition 11, we can easily formulate the natural partition problems that were stated at the beginning of Section 3, where in all cases we use $t = 2$.

Biclique: Here we use $L^{\texttt{rel}}_{1,2} = 1$ and $H^{\texttt{abs}}_{1,1} = H^{\texttt{abs}}_{2,2} = 0$.

That is, we mandate maximal edge density between the two parts (i.e., no edges may be missing) and minimal edge density within each part (i.e., no edges may be present there).

All other parameters are trivial, which means that the lower bounds (e.g., L_i's) are all set to 0, while the upper bounds (e.g., H_i's) are all set to 1.

[27] Indeed, Condition 4 could have been integrated in Condition 3 if we had fictitiously defined $E(V', V')$ to include self-loops and two copies of each edge.

Bipartiteness: Here we use $H_{1,1}^{\mathsf{abs}} = H_{2,2}^{\mathsf{abs}} = 0$. Again, all other parameters are trivial.

Max-Cut (for $\rho \in (0, 0.25]$): Here we use $L_{1,2}^{\mathsf{abs}} = \rho$ (and again all other parameters are trivial).

Min-Bisection (for $\rho \in (0, 0.25]$): Here we use $H_{1,2}^{\mathsf{abs}} = \rho$ and $L_1 = L_2 = H_1 = H_2 = 1/2$.

Max-Clique (for $\rho \in (01]$): Here we use $L_i = \rho$ and $L_{1,1}^{\mathtt{rel}} = 1$.

The following result follows from the techniques used in the proof of [26, Thm. 9.1].[28]

Theorem 12 (testing general partition problems (in the dense graph model)): *Every graph partition problem can be ϵ-tested within query complexity* $\text{poly}(t/\epsilon)^t$, *where the polynomial does not depend on the parameters of the graph partition problem and t is the first parameter of the problem* (cf., Definition 11). *The computational complexity of the tester is exponential in its query complexity.*

The tester operates by selecting a sample of $\text{poly}(t/\epsilon)^t$ vertices and checking whether the induced subgraph satisfies the very same graph partition problem, possibly up to a small relaxation in the density parameters.[29] The latter checking is done by merely going over all possible t-partitions of the induced graph and checking if any of them satisfies the corresponding property. This explains the exponential time bound, which seems unavoidable in general, because a $T(1/\epsilon)$ time bound for ϵ-testing properties such as Max-Cut or 3-coloring would have implied a $T(k^2)$-time algorithm for these problems (by setting $\epsilon = 1/k^2$).

Finding approximately good partitions. As in the case of `Bipartiteness`, the tester for each graph partition problem can be modified into an algorithm that finds an (succinct representation of an) approximately adequate partition whenever it exists. That is, if the k-vertex graph has the desired (t-partition) property, then the testing algorithm may actually output auxiliary information that allows to reconstruct, in $\text{poly}(1/\epsilon) \cdot k$-time, a t-partition that approximately obeys the property. (For example, for ρ-`Cut`, we can construct a 2-partition with at least $(\rho - \epsilon) \cdot k^2$ crossing edges.) Furthermore, the location of each vertex with respect to that t-partition can be determined in $\text{poly}(1/\epsilon)$-time. We comment that this notion of a succinct representation of a structure that corresponds to an (approximate) NP-witness may be relevant for other sets in $\mathcal{NP}$ (i.e., not only to graph partition problems).[30]

[28] Indeed, [26, Thm. 9.1] only refers to the case in which all the relative bounds (i.e., the $L_{i,j}^{\mathtt{rel}}$'s and $H_{i,j}^{\mathtt{rel}}$'s) are trivial, since such bounds were not included in the definition used in [26, Sec. 9]. Nevertheless, the proof seems to extend in a straightforward manner, if one can use such an expression when referring to such a complex proof. Verifying this belief and providing a detailed proof would be a worthy project.

[29] The analysis of the tester uses the force-and-check technique. In particular, we consider all possible t-partitions of the first part of the sample, denoted U, as well as all possible (approximate) values for a sequence of auxiliary parameters. Each such pair of choices induces a t-partition of $[k]$. It is shown that if the input graph satisfies the property, then one of these t-partitions of $[k]$ witnesses this fact, and that it is possible to determine the location of every vertex that is adjacent to U with respect each of these partitions based on its adjacency relation with U (and the auxiliary parameters), where all but at most $0.1\epsilon k^2$ of the edges are adjacent to vertices in $\Gamma(U)$. The details are quite tedious, but this is a merely complex incarnation of the abstract outline that followed the proof of Proposition 6.

[30] Indeed, an interesting algorithmic application was presented in [20], where an implicit partition of an imaginary hypergraph is used in order to efficiently construct a regular partition (with almost optimal parameters) of a given graph.

The case of t-`Colorability`. We mention that better bounds are known for some specific properties that fall into the framework of Definition 11. Most notably, t-`Colorability` (i.e., $H_{i,i}^{\mathsf{abs}} = 0$ for all $i \in [t]$) can be ϵ-tested using $\mathrm{poly}(t/\epsilon)$ queries. In this case, the tester selects a random sample of $\widetilde{O}(t/\epsilon^2)$ vertices and accepts if and only if the induced subgraph is t-colorable. Recall that for `2-Colorability` (i.e., `Bipartiteness`), a random sample of $\widetilde{O}(1/\epsilon)$ vertices suffices. Let us state these results in combinatorial terms.

Theorem 13 (testing t-`Colorability` (in the dense graph model)):[31] *For every $t \geq 2$, if a graph G is ϵ-far from being t-colorable, then, with high probability, a random induced subgraph of size $\widetilde{O}(t/\epsilon^{c_t})$ of G is not t-colorable, where $c_2 = 1$ and $c_t = 2$ otherwise.*

4 Connection to Szemeŕedi's Regularity Lemma

The problem of testing graph properties (in the dense graph model) is related to Szemeŕedi's Regularity Lemma [46]. This relation arises when focusing on the question of *which graph properties are testable within query complexity that only depends on the proximity parameter?*

We stress the fact that the foregoing question ignores the specific dependence (of the query complexity on the proximity parameter). It rather stresses the independence of the query complexity from the size of the graph, and it seems adequate to say that such properties have size-oblivious tester, although this term is a bit misleading (since the tester must use the size parameter in order to operate).[32]

4.1 The Regularity Lemma

Recall that for a graph $G = (V, E)$ and two disjoint sets $A, B \subseteq V$, we denote by $E(A, B)$ the set of edges having one endpoint in A and another endpoint in B.

Definition 14 (edge density and regular pairs): *Let $G = (V, E)$ be a graph and $A, B \subseteq V$ be disjoint and non-empty sets of vertices.*

- *The* edge density of the pair (A, B) *is defined as* $d(A, B) \stackrel{\text{def}}{=} \frac{|E(A,B)|}{|A|\cdot|B|}$.
- *The pair (A, B) is said to be* γ-regular *if for every $A' \subseteq A$ and $B' \subseteq B$ such that $|A'| \geq \gamma \cdot |A|$ and $|B'| \geq \gamma \cdot |B|$ it holds that $|d(A', B') - d(A, B)| \leq \gamma$.*

In many ways, a regular pair in a graph "looks like" a random bipartite graph of the some edge density; that is, the reader may think of and analyze a regular pair as if it was such a random bipartite graph, and the conclusion reached in such an analysis would typically hold for the regular pair.[33] Indeed, for sufficiently large A and B, a random bipartite graph between A and B is regular

[31] Note that the problem of 1-coloring is almost trivial, since it asks if the graph is empty.

[32] For starters, even selecting a uniformly distributed vertex requires knowing the number of vertices. In addition, as pointed out by Alon and Shapira [9], the final decision of the tester may also depend on the number of vertices. A trivial example refers to the graph property that requires having an odd number of vertices. In any case, the term "size-oblivious testability" seems much better than the term "testability" which is often used (when referring to the independence of the query complexity from the size of the graph).

[33] Of course, the word "typically" is crucial here, and it refers to natural assertions that one may want to make on graphs. For example, if the regular pair (A, B) has edge density ρ, then almost all vertices in A have degree that is approximately $\rho \cdot |B|$, and almost all pairs of vertices in A have approximately $\rho^2 \cdot |B|$ common neighbors in B. See Exercise 7.

with very high probability (see Exercise 8). The regularity lemma asserts that, for every $\ell \in \mathbb{N}$ and $\gamma > 0$, every sufficiently large graph can be partitioned into (at least ℓ) almost equal sets such that all but at most a γ fraction of the set-pairs are γ-regular, where the number of parts is upper-bounded by a function of ℓ and γ. That is:

Theorem 15 (Szemeŕedi's Regularity Lemma [46]):[34] *For every $\ell \in \mathbb{N}$ and $\gamma > 0$ there exists a $T = T(\ell, \gamma)$ such that every sufficiently large graph $G = (V, E)$ there exists a $t \in [\ell, T]$ and a t-partition of V, denoted $(V_1, ..., V_t)$ that satisfies the following two conditions:*

1. Equipartition: *For every $i \in [t]$, it holds that $\lfloor |V|/t \rfloor \leq |V_i| \leq \lceil |V|/t \rceil$.*
2. Regularity: *For all but at most a γ fraction of the pairs $\{i, j\} \in \binom{[t]}{2}$, it holds that (V_i, V_j) is γ-regular.*

Intuitively this means that every graph graph can be equipartitioned into a constant number of parts such that almost all pairs of parts looks like a random bipartite graph of the some edge density. The said constant depends on the parameters ℓ and γ, alas the bound for this quantity (i.e., $T(\ell, \gamma)$) is a tower of poly$(1/\gamma)$ exponents; that is, $T(\ell, \gamma) = \mathtt{T}(\text{poly}(1/\gamma))$, where $\mathtt{T}$ is defined inductively by $\mathtt{T}(m) = \exp(\mathtt{T}(m-1))$ with $\mathtt{T}(1) = 2$. It turns out that this huge upper bound cannot be significantly improved, since $T(\ell, \gamma) = \mathtt{T}((1/\gamma)^{\Omega(1)})$ is a lower bound on the number of required sets [36]. (A proof of Theorem 15 can be found in many sources; see, for example, [10, Sec. 9.4].)[35]

4.2 Subgraph freeness

The relevance of the regularity lemma to property testing can be illustrated by considering the problem of testing *H-freeness*, for a fixed graph H (say the triangle).

Definition 16 (subgraph freeness): *Let H be a fixed graph. A graph $G = (V, E)$ is H-*free *if G contains no subgraph that is isomorphic to H.*[36]

(For example, if H contains a single edge, then H-freeness means having no edges.)[37] We stress that Definition 16 requires that G contains no copy of H as a subgraph, and this is a more strict requirement than requiring that G contains no *induced subgraph* that is isomorphic to H. (The difference between these two notion arises when H is not a clique.)

[34] An alternative (popular) formulation requires all sets to be of equal size, but allows an exceptional set of size at most $\gamma \cdot |V|$.

[35] The basic idea is to start with an arbitrary ℓ-equipartition and "refine" it in iteration till the current partition satisfies the regularity condition. If the current t-partition violates the regularity condition, then the $\gamma \cdot \binom{t}{2}$ non-regular pairs give rise to a 2^t-partition of each of the original parts such that some potential function, which ranges in $[0, 1]$, increases by at least poly(γ). This yields a refinement of the original t-partition, which yields a $\exp(O(t))$-equipartition (by further refinement, which never decreases the potential). Hence we have poly(γ) many refinement steps, where in each step the number of parts grows exponentially. Finally, we mention that the potential function used assigns the partition $(V_1, ..., V_t)$ of $[k]$, the value $\sum_{i<j} \frac{|V_i| \cdot |V_j|}{k^2} \cdot d(V_i, V_j)^2$. The verification of the aforementioned features of this potential function is left to Exercise 9.

[36] That is, if $H = ([t], F)$, then G is H-free if and only if for every one-to-one mapping $\phi : [t] \rightarrow V$ there exists an edge $\{i, j\} \in F$ such that $\{\phi(i), \phi(j)\} \notin E$. Equivalently, G is *not* H-free if and only if there exists a one-to-one mapping $\phi : [t] \rightarrow V$ such that $\{\{\pi(i), \pi(j)\} : \{i, j\} \in F\} \subseteq E$.

[37] Hence, our focus is on graphs H that have at least two edges, which means that they have at least three vertices.

Suppose that H is a t-vertex graph. Then, a natural (one-sided error) proximity oblivious tester for H-freeness consists of selecting t random vertices in the tested graph, and checking whether the induced subgraph contains a copy of H. The question is what is the rejection probability of this (one-sided error) tester. In other words, we pose the following question (for which only partial answers, reviewed next, are known).[38]

Open Problem 17 (the number of copies of H in graphs that are ϵ-far from H-free): *Let H be a connected t-vertex graph and let $\#_H(\epsilon, k)$ denote the minimal number of copies of H in a k-vertex graph that is ϵ-far from being H-free. Provide relatively tight lower and upper bounds on $\#_H(\epsilon, k)$.*

Note that, for $t \geq 3$, it is not a priori clear whether $\#_H(\epsilon, k)$ can be lower bounded by $\rho_H(\epsilon) \cdot \binom{k}{t}$ for any function $\rho_H : (0, 1] \to \mathbb{N}$. Such a lower bound is established using the Regularity Lemma, and no other proof of it is know when H is *not* bipartite. Furthermore, for any non-bipartite t-vertex graph H, the known bounds on the function ρ_H are far apart. Interestingly, in this case it is known that $\rho_H(\epsilon) \ll \text{poly}(\epsilon)$. For example, if G is ϵ-far from triangle-free, then it does *not* follow that G has $\text{poly}(\epsilon) \cdot \binom{k}{3}$ triangles. These striking facts are summarized in the following theorem.

Theorem 18 (upper and lower bounds on $\#_H(\epsilon, k)$): *Let H and $\#_H(\epsilon, k)$ be as in Problem 17. Then, it holds that*

1. *$\#_H(\epsilon, k) \geq \rho_H((0.1\epsilon)^{t-2}/t)^t \cdot \binom{k}{t}$ for $\rho_H(\epsilon')$ that is the reciprocal of a tower of* $\text{poly}(1/\epsilon')$ *exponents.*
2. *If H is bipartite, then $\#_H(\epsilon, k) \geq \rho_H(\epsilon) \cdot \binom{k}{t}$ for $\rho_H(\epsilon) = \Omega(\epsilon^{t^2/4})$.*
3. *If H is not bipartite, then for every positive polynomial p it holds that $\#_H(\epsilon, k) < p(\epsilon) \cdot \binom{k}{t}$. In fact, $\#_H(\epsilon, k) < \exp(-\Omega(\log(1/\epsilon))^2) \cdot \binom{k}{t}$.*

(Recall that t denotes the number of vertices in H.)

Theorem 18 summarizes the state of knowledge with respect to Problem 17, and indeed it leaves much to be understood (i.e., note the huge gap between Parts 1 and 3). Nevertheless, Theorem 18 suffices for establishing the existence of (one-sided error) proximity oblivious tester for all subgraph-freeness properties. Specifically, H-freeness has a proximity oblivious tester with detection probability function ρ_H as asserted in Parts 1 and 2, but for non-bipartite H this detection probability is *not* polynomial in the distance from the corresponding property. Furthermore, in that case, H-freeness has no ϵ-tester of $\text{poly}(1/\epsilon)$ query complexity, even when allowing two-sided error [5].[39] Here we shall only prove Part 1; the proofs of Parts 2 and 3 can be found in [1].[40]

[38] We focus on the case of connected t-vertex graphs H, while noting that the general case is reducible to it. Specifically, if G is ϵ-far from being H-free and H' is a connected component of H, then G is ϵ-far from being H'-free. Hence, if for every connected t'-vertex graph H' it holds that $\#_{H'}(\epsilon, k) \geq \rho_{H'}(\epsilon) \cdot \binom{k}{t'}$ for some function $\rho_{H'} : (0, 1] \to \mathbb{N}$, then the same holds for unconnected graphs H, because the number of intersections between copies of different connected components of H is at least one order of magnitude smaller: Specifically, the number of copies of H' that intersect copies of a t''-vertex H'' is at most $t't'' \cdot \binom{k}{t'+t''-1}$, whereas the number of $(t' + t'')$-vertex sets that contain copies of both H' and H'' is at least $\rho_{H'}(\epsilon) \cdot \binom{k}{t'} \cdot \rho_{H''}(\epsilon) \cdot \binom{k}{t''}$.

[39] For induced subgraph freeness, this lower bound holds for any graph H that has at least five vertices, regardless if it is bipartite or not [8].

[40] The proof of Part 2 reduces to the fact (cf. [1, Lem. 2.1] or Exercise 10) that if a k-vertex graph has at least ϵk^2 edges, then it contains at least $\Omega((2\epsilon)^{t_1 t_2}) \cdot k^{t_1+t_2}$ copies of K_{t_1,t_2} (i.e., the biclique with t_1 vertices on one side and

Proof of Part 1: Fixing any k-vertex graph $G = ([k], E)$ that is ϵ-far from being H-free, we set $\gamma = (0.1\epsilon)^{t-2}$ and $\ell = 10/\epsilon$, and apply the regularity lemma G. Denoting the partition provided by the regularity lemma, by $(V_1, ..., V_T)$, where T is upper-bounded by a tower of poly$(1/\gamma)$ exponents, we modify G as follows:

1. We omit all edges that are internal to any of the V_i's.

   In total, we omitted at most $T \cdot \binom{\lceil k/T \rceil}{2} < k^2/T \leq k^2/\ell = 0.1\epsilon k^2$ edges.

2. We omit all edges between pairs of sets that are not γ-regular.

   Here, we omitted at most $\gamma \cdot \binom{T}{2} \cdot \lceil k/T \rceil^2 < \gamma \cdot k^2 \leq 0.1\epsilon k^2$ edges.

3. We omit all edges between pairs of sets that have edge density below 0.2ϵ; that is, we omit all edges between V_i and V_j if and only if $d(V_i, V_j) \leq 0.2\epsilon$.

   Here, we omitted at most $\binom{T}{2} \cdot 0.2\epsilon \cdot \lceil k/T \rceil^2 < 0.1\epsilon \cdot k^2$ edges.

Hence, the resulting graph, denoted $G' = ([k], E')$, is a subgraph of G that is *not* H-free.[41] Furthermore, by Steps 2 and 3, every pair (V_i, V_j) is γ-regular in G' and has edge density that is either at least 0.2ϵ or equals zero (i.e., there are no edges between V_i and V_j in G'). Lastly, by Step 1, the graph G' contains no edges that are internal to any V_i.

Given that G' contains some copies of H, we shall lower bound the number of copies of H in G'. At this point we invoke the intuition provided right after Definition 14, by which regular pairs behave as random bipartite graphs of similar edge density. Considering the guaranteed copy of $H = ([t], F)$ in $G' = ([k], E')$, we observe that its edges reside in regular pairs that have edge density at least 0.2ϵ. If these regular pairs would behave as random bipartite graphs of similar density, then we should expect to have at least $(0.2\epsilon)^{|F|} \cdot (k/T)^t$ copies of H on G', due merely to these regular pairs, and the Part 1 would follow (since this quantity is $\Omega(\epsilon)^{t^2} \cdot T^{-t} \cdot \binom{k}{t})$). The actual proof amounts to materializing this observation in the real setting in which the regular pairs are fixed bipartite graphs rather than being random bipartite graphs of similar densities.

Starting the actual proof and considering the guaranteed copy of H in G', we make the following initial observations. We first observe that if H is a clique, then this copy (of H) contains at most one vertex in each of the V_i's, since each pair of vertices in the copy of H must be connected in G' (whereas vertices in the same V_i are not connected). Turning to the general case (i.e., a general t-vertex graph H), we admit that a copy of H may contain several vertices in the same V_i. But, in such a case, we can partition each V_i into t equal parts, while noting that the regularity condition is preserved, except that the regularity parameter is now t times bigger.[42] Hence, we should actually invoke the regularity lemma with $\gamma = (0.1\epsilon)^{t-2}/t$ (rather than with $\gamma = (0.1\epsilon)^{t-2}$). We shall assume, without loss of generality, that the i^{th} vertex of the foregoing copy of H resides in V_i. Furthermore, we observe that if V_i and V_j contain vertices of this copy (of H) that are

t_2 vertices on the other side). Hence, if the k-vertex graph G is ϵ-far from being H-free, then G must be ϵ-far from the empty graph, and hence contain at least $\Omega((2\epsilon)^{t'\cdot(t-t')}) \cdot k^t$ copies of $K_{t',t-t'}$ for every $t' \in [t-1]$. (Thus, if H is a subgraph of $K_{t',t-t'}$, then G contain at least $\Omega((2\epsilon)^{t'\cdot(t-t')}) \cdot k^t$ copies of H.) We also mention that a two-sided error ϵ-tester of query complexity $O(1/\epsilon)$ (for H-freeness) can just estimate the number of edges in the tested graph, and reject if and only if it is safe to say that the graph has more than $0.4\epsilon k^2$ edges (cf., Proposition 4). A partial proof of Part 3 can be found in [45, Sec. 9.1].

[41] Indeed, although we can show that G' is 0.4ϵ-far from being H-free, we only use the fact that G' is not H-free.

[42] Since every (V_i, V_j) is γ-regular, each of the t^2 resulting pairs is $t\gamma$-regular (see Exerecise 11). Also, since there are no edges between vertices of V_i there will be no edges between its t parts.

connected in H, then (by Steps 2 and 3) the pair (V_i, V_j) is γ-regular and has edge density at least 0.2ϵ. Let us summarize:

***Starting point*:** The graph G' contains a copy of H such that the i^{th} vertex of the foregoing copy of H resides in V_i, and if i and j are connected in H then the pair (V_i, V_j) has edge density at least 0.2ϵ (and is γ-regular).

We now consider an auxiliary graph $A = ([T], F)$ such that $\{i, j\} \in F$ if and only if there is an edge in G' between some vertex of V_i and some vertex of V_j (i.e., there exists $u \in V_i$ and $v \in V_j$ such that $\{u, v\} \in E'$). However, according to Step 3, the existence of a single edge (in G') between V_i and V_j implies the existence of at least $0.2\epsilon \cdot \lfloor k/T \rfloor^2$ such edges. Furthermore, by Steps 2 and 3, if $\{i, j_1\}, \{i, j_2\} \in F$ (equivalently, if there are edges (in G') between V_i and both V_{j_1} and V_{j_2}), then there are many vertices in V_i that have many edges to both V_{j_1} and V_{j_2} (in G'). A more elaborate argument, which is presented next, shows that the existence of any t-vertex subgraph in A, implies that this subgraph appears in "abundance" in G'. This fact combined with the fact that A must contain a copy of H (since G' is not H-free), implies that G' (and so also G) conatins many copies of H. Let us first detail the argument for the case that H is the (three-vertex) triangle.

The case in which H is a triangle. Since the graph G' is not triangle-free, it follows that the graph A contains a triangle (which, w.l.o.g, consists of the vertices $1, 2$ and 3). Turning back to G', for each vertex $v \in V_1$, we consider its neighbors in V_2 and V_3, and denote the corresponding sets by $\Gamma_2(v)$ and $\Gamma_3(v)$, respectively; that is, $\Gamma_i(v) = \{u \in V_i : \{u, v\} \in E'\}$. We make the following two observations:

***Observation 1*:** If $|\Gamma_i(v)| \geq 0.1\epsilon \cdot |V_i|$ for both $i \in \{2, 3\}$, then the number of triangles that involve v is at least $(0.1\epsilon)^3 \cdot \lfloor k/T \rfloor^2$.

This follows since for such a vertex v, each pair $(w_2, w_3) \in \Gamma_2(v) \times \Gamma_3(v)$ such that $\{w_2, w_3\} \in E'$ yields a triangle, whereas the density of such edges (i.e., edges between $\Gamma_2(v)$ and $\Gamma_3(v)$) is approximately the density of edges between V_2 and V_3. Specifically, we have

$$\begin{aligned} d_{G'}(\Gamma_2(v), \Gamma_3(v)) \cdot |\Gamma_2(v)| \cdot |\Gamma_3(v)| &\geq (d_{G'}(V_2, V_3) - \gamma) \cdot |\Gamma_2(v)| \cdot |\Gamma_3(v)| \\ &\geq 0.1\epsilon \cdot (0.1\epsilon \cdot \lfloor k/T \rfloor)^2 \end{aligned}$$

where the first inequality uses the fact that (V_2, V_3) is a γ-regular pair (and $|\Gamma_i(v)| \geq 0.1\epsilon \cdot |V_i|$ for both $i \in \{2, 3\}$), whereas the second inequality uses the fact that (V_2, V_3) has edge density at least 0.2ϵ (and $\gamma \leq 0.1\epsilon$).

***Observation 2*:** Most of the vertices $v \in V_1$ satisfy $|\Gamma_i(v)| \geq 0.1\epsilon \cdot |V_i|$ for both $i \in \{2, 3\}$. In fact, for every $i \in \{1, 2\}$, at least a $1 - \gamma$ fraction of the vertices $v \in V_1$ satisfy $|\Gamma_i(v)| \geq 0.1\epsilon \cdot |V_i|$.

To see this, let $V_1' \stackrel{\text{def}}{=} \{v \in V_1 : |\Gamma_i(v)| < 0.1\epsilon \cdot |V_i|\}$, and assume towards the contradiction that $|V_1'| > \gamma \cdot |V_1|$. Now, since the pair (V_1, V_i) is γ-regular (and $|V_1'| \geq \gamma \cdot |V_1|$), we have

$$\begin{aligned} d_{G'}(V_1', V_i) \cdot |V_1'| \cdot |V_i| &\geq (d_{G'}(V_1, V_i) - \gamma) \cdot |V_1'| \cdot |V_i| \\ &\geq 0.1\epsilon \cdot |V_1'| \cdot |V_i| \end{aligned}$$

but this contradicts the definition of V_1', which asserts that each $v \in V_1'$ has less than $0.1\epsilon \cdot |V_i|$ neighbors in V_i.

Combining the two observations, we conclude that there are at least $0.5|V_1| \cdot (0.1\epsilon)^3 \cdot \lfloor k/T \rfloor^2 > 0.4 \cdot (0.1\epsilon/T)^3 \cdot k^3$ triangles in G'. Recalling that T is upper-bounded by function of ϵ (i.e., a tower of $\mathrm{poly}(1/(0.1\epsilon))$ exponents), Part 1 follows in this case (in which H is a triangle).

The general case: arbitrary H. We now turn to the general case in which H is an arbitrary t-vertex graph. Recall that by our hypothesis G' contains a copy of H with a single vertex in V_i for every $i \in [t]$. It follows that the auxiliary graph A contains a copy of H, and that this copy resides on the vertices $1, 2, ..., t$. In this case we proceed in $t-2$ iterations, starting with $H^{(0)} = H$ and $V_i^{(0)} = V_i$ for every $i \in [t]$.

In the i^{th} iteration, we claim that, *at least half of the vertices in* $V_i^{(i-1)}$ *have at least* $0.1\epsilon \cdot |V_j^{(i-1)}|$ *neighbors in* $V_j^{(i-1)}$ *for every* $j > i$ *that is a neighbor of* i *in* $H^{(i-1)}$. This claim is analogous to Observation 2 (above), and is proved in the same manner (while relying on $|V_i^{(i-1)}| \geq \gamma \cdot |V_i|$, which will be established below). For each such vertex v, we define $V_j^{(i)} = V_j^{(i-1)}$ if j does not neighbor i in $H^{(i-1)}$ and $V_j^{(i)} = \{u \in V_j^{(i-1)} : \{u, v\} \in E'\}$ otherwise.[43] We then argue as follows:

1. Let $H^{(i)}$ be the subgraph of $H^{(i-1)}$ induced by $\{i+1, ..., t\}$. Then, *the number of copies of* $H^{(i-1)}$ *in* G' *that involve* v *as well as a single vertex from each* $V_j^{(i-1)}$ *for* $j \in \{i+1, ..., t\}$ *is lower-bounded by the number of copies of* $H^{(i)}$ *in* G' *that contain a single vertex from each* $V_j^{(i)}$ *for* $j \in \{i+1, ..., t\}$.

2. For every $j \in \{i+1, ..., t\}$, it holds that $|V_j^{(i)}| \geq 0.1\epsilon \cdot |V_j^{(i-1)}| \geq (0.1\epsilon)^i \cdot |V_j|$. Hence, for every $i \in [t-2]$ and $j \in \{i+1, ..., t\}$, it holds that $|V_j^{(i)}| \geq \gamma \cdot |V_j|$. (In particular, $|V_{i+1}^{(i)}| \geq \gamma \cdot |V_{i+1}|$.)

It follows that the number of copies of $H^{(i-1)}$ in G' that involve a single vertex from each $V_j^{(i)}$ for $j \in \{i, ..., t\}$ is at least $0.5 \cdot |V_i^{(i-1)}| \geq 0.5 \cdot (0.1\epsilon)^{i-1} \cdot |V_i|$ times the number of copies of $H^{(i)}$ in G' that involve a single vertex from each $V_j^{(i)}$ for $j \in \{i+1, ..., t\}$.[44] Lastly, we show that the number of copies of $H^{(t-2)}$ in G' that involve a single vertex from each $V_j^{(t-2)}$ for $j \in \{t-1, t\}$ is at least $0.1\epsilon \cdot |V_{t-1}^{(t-2)}| \cdot |V_t^{(t-2)}|$, which is at least $0.1\epsilon \cdot ((0.1\epsilon)^{t-2} \cdot \lfloor k/t \rfloor)^2$. This claim is analogous to Observation 1 (above), and is proved in the same manner (while relying on $|V_j^{(t-2)}| \geq \gamma |V_j|$ for both $j \in \{t-1, t\}$).[45] Hence, the number of copies of H in G' is at least

$$\prod_{i=1}^{t-2} \left(0.5 \cdot (0.1\epsilon)^{i-1} \cdot |V_i|\right) \; \cdot \; 0.1\epsilon \cdot \left((0.1\epsilon)^{t-2} \cdot \lfloor k/t \rfloor\right)^2$$

[43] The notation $V_j^{(i)}$ is imprecise, since this set depends on v as well as the sequence of vertices fixed in the prior $i-1$ iterations. That is, for every choice of $(v_1, .., v_{i-1}) \in V_1 \times \cdots \times V_{i-1}$ made in the prior $i-1$ iterations, we claim that (for every $j > i$ that is a neighbor of i in $H^{(i-1)}$), it holds that at least half of the vertices in $V_i^{(v_1, ..., v_{i-1})}$ have at least $0.1\epsilon \cdot |V_j^{(v_1, ..., v_{i-1})}|$ neighbors in $V_j^{(v_1, ..., v_{i-1})}$. For every such vertex v_i, we define $V_j^{(v_1, ..., v_i)} = V_j^{(v_1, ..., v_{i-1})}$ if j does not neighbor i in $H^{(i-1)}$ and $V_j^{(v_1, ..., v_i)} = \{u \in V_j^{(v_1, ..., v_{i-1})} : \{u, v_i\} \in E'\}$ otherwise.

[44] Recall that for most vertices $v \in V_i^{(i-1)}$, it holds that that number of copies of $H^{(i-1)}$ in which v participates equals the number of copies of $H^{(i)}$ in G' that involve a single vertex from each $V_j^{(i)}$ for $j \in \{i+1, ..., t\}$, where the $V_j^{(i)}$'s are defined based on v and $H^{(i-1)}$.

[45] Alternatively, we can use yet another iteration, while setting $\gamma = (0.1\epsilon)^{t-1}$ (rather than $\gamma = (0.1\epsilon)^{t-2}$), and use the corresponding claim regarding $H^{(t-1)}$, which is trivial.

$$
\begin{aligned}
&> \left((0.1\epsilon)^{t-2} \cdot \lfloor k/T \rfloor\right)^{t-2} \cdot (0.1\epsilon)^{2t-3} \cdot \lfloor k/T \rfloor^2 \\
&> \frac{(0.1\epsilon)^{t^2}}{T^t} \cdot k^t
\end{aligned}
$$

and the claim follows. ■

Digest: On an apparent waste in the proof. The reader may wonder why we did not use the fact that G' is actually 0.4ϵ-far from being H-free (rather than only using the fact that G' is not H-free). Using this stronger fact, we can indeed infer that the auxiliary graph A is 0.4ϵ-far from being H-free. But we cannot capitalize on the latter fact, since we do not have a good lower bound on the number of copies of H in A. Indeed, getting such a lower bound is the contents of Part 1 of Theorem 18, but the result established there is meaningless for graphs of size T (such as A). The only lower bound that is obvious with respect to A, is that (the T-vertex graph) A has at least $\Omega(\epsilon T^2/t)$ different t-vertex subsets that contain a copy of H, since omitting all edges that are incident at any such vertex (in any such t-subset) eliminates all copies of H.[46] But, at best, this will only allow us to assert that $\#_H(\epsilon, k) \geq \Omega(\epsilon T^2) \cdot (k/T)^t$, which is not significantly better than the bound $\#_H(\epsilon, k) \geq (0.1\epsilon/T)^t \cdot \binom{k}{t}$ that we just proved.

Summary. For sake of good order, we spell out the results regarding testing subgraph freeness that are implied by Theorem 18 (and by the discussion that followed it (including Footnote 40)).

Corollary 19 (on the complexity of testing subgraph freeness (in the dense graph model)): *Let H be a t-vertex graph. Then:*

1. *There exists a one-sided error proximity oblivious tester that makes t queries and has detection probability $\varrho_H(\delta) = 1/\mathtt{T}(\mathrm{poly}(\delta^t/t))^t$, where $\mathtt{T}$ is the tower-of-exponents function* (i.e., $\mathtt{T}(m) = \exp(\mathtt{T}(m-1))$ and $\mathtt{T}(1) = 2$).
2. *If H is bipartite, then there exists a one-sided error proximity oblivious tester that makes t queries and has detection probability $\varrho_H(\delta) = \Omega(\delta^{t^2/4})$. In this case, H-freeness has a two-sided error tester of query complexity $O(1/\epsilon)$.*[47]
3. *If H is not bipartite, then H-freeness has no ϵ-tester of* $\mathrm{poly}(1/\epsilon)$ *query complexity, even when allowing two-sided error.*[48]

We mention that the corresponding properties that refer to *induced subgraphs freeness* also have constant-query (one-sided error) proximity oblivious testers, but their detection probability is even worse (i.e., it is a tower of tower functions [8]).[49] Furthermore, this result extends to the case that the property postulates freeness for a family of graphs; that is, for a fixed family of (forbidden)

[46] Note that the subgraph induced by each t-subset may contain several different copies of H, but since H is connected it suffices to disconnect one of the vertices in the t-subset from all other vertices in this subset.

[47] See Footnote 40.

[48] Indeed, this result (of Alon and Shapira [5]) is stronger than the corresponding part of Theorem 18: It refers to general testers (rather than to one-sided error testers that arise from repeating a t-query proximity oblivious tester for a predetermined number of times.

[49] Recall that a graph G is H-free if G contains no subgraph that is isomorphic to H. In contrast, G is induced H-free if G contains no *induced* subgraph that is isomorphic to H.

graphs $\mathcal{H}$, a graph G is induced $\mathcal{H}$-free if G contains no induced subgraph that is isomorphic to a graph in $\mathcal{H}$. (Note that here we focus on induced subgraph freeness, since non-induced subgraph freeness with respect to a finite set of graphs $\mathcal{H}$, can be captured by induced subgraph freeness with respect to a finite set of graphs $\mathcal{H}'$.)[50]

Actually, the foregoing result (i.e., that every induced subgraph freeness property has a constant-query proximity oblivious tester) is, in some sense, the strongest possible. Loosely speaking, *a graph property has a constant-query* (one-sided error) *proximity-oblivious tester if and only if it expressible as an induced subgraph freeness property.* Recall that a proximity-oblivious tester (POT) is required to have detection probability that only depends on the distance of the tested object from the property. The actual result, stated next, allows the family of forbidden subgraphs to depend on the number of vertices in the tested graph, as long as the number of vertices in each graph in the family is uniformly bounded.

Theorem 20 (characterization of graph properties having a POT (in the dense graph model)): *Let $\Pi = \bigcup_{k\in\mathbb{N}} \Pi_k$ be a graph property such that each Π_k consists of all k-vertex graphs that satisfy Π. Then, Π has a constant-query* (one-sided error) *proximity-oblivious tester if and only if there exist a constant c and an infinite sequence $\overline{\mathcal{H}} = (\mathcal{H}_k)_{k\in\mathbb{N}}$ of sets of graphs such that*

1. *each $\mathcal{H}_k$ contains graphs of size at most c; and*
2. *Π_k equals the set of k-vertex graphs that are induced $\mathcal{H}_k$-free.*

(Note that the number of possible $\mathcal{H}_k$'s is upper bounded by a function of c; indeed, it is at most double-exponential in c^2.)[51] The existence of POTs for properties that satisfy the (induced subgraph) condition follows from [8], whereas the opposite direction is based on Theorem 25 (below).

4.3 The structure of properties that have size-oblivious testers

The role of the regularity lemma is not confined to proving the existence of proximity-oblivious testers for any graph property that is expressible as an induced subgraph freeness property. It turns out that every graph property that can be tested using a number of queries that is independent of the size of the graph can be expressed in terms properties having a regular partition that fits a given sequence of edge densities. (In the following definition, t denotes the number of parts is the partition, γ denotes the regularity parameter, C denotes the set of regular pairs, and the $d_{i,j}$'s denote the prescribed densities.)

Definition 21 (regularity properties): *A* regularity property *is parameterized by a sequence*

$$(\gamma, t, (d_{i,j})_{i<j:i,j\in[t]}, C)$$

such that $\gamma \in (0,1]$ and $C \subset \binom{[t]}{2}$ has size $\lceil (1-\gamma)\cdot\binom{t}{2}\rceil$. This property consists of all graphs $G=(V,E)$ such that there exists a t-equipartition of V, denoted $(V_1, ..., V_t)$, and for every $(i,j)\in C$ the pair (V_i, V_j) is γ-regular and $|E(V_i,V_j)| = \lfloor d_{i,j}\cdot|V_i|\cdot|V_j|\rfloor$. We call $\max(\gamma, 1/t)$ the fineness *of the property.*

[50] Specifically, suppose that $\mathcal{H}$ contains graphs with at most t vertices. Then, $\mathcal{H}'$ is the set set of all t-vertex graphs that contain a subgraphs that is in $\mathcal{H}$. Note that G contains a (general) subgraph that is isomorphic to a graph in $\mathcal{H}'$ if and only if G contains an induced subgraph that is isomorphic to a graph in $\mathcal{H}$.

[51] This fact is important towards applying the result of [8], which relates to the case that $\mathcal{H}_k$ is independent of k. Note that a property Π that satisfies the "$\overline{\mathcal{H}}$-freeness" condition is a union of a finite number of (trivially modified) induced freeness properties (as in [8]).

We shall consider properties that can be expressed as the union of a finite number of regularity properties of a bounded fineness. In fact, we shall refer to properties that are approximated by the latter, where the notion of approximation is as defined in the notes on testing by implicit sampling.

Definition 22 (approximation of a property): *The property Π is δ-approximated by the property Π' if each object in Π is δ-close to some object in Π', and vice versa.*

We are finally ready to state the result alluded to above. It asserts that every graph property that can be tested using a number of queries that is independent of the size of the graph can be approximated by the union of regularity properties (where fineness of these properties is lower-bounded in terms of the approximation parameter). Actually, the converse holds as well.

Theorem 23 (characterization of properties that have size-oblivious testers (in the dense graph model)): *Let Π be a graph property. Then, the following two conditions are equivalent.*

1. *There exists a function $q : (0,1] \rightarrow \mathbb{N}$ such that the property Π has a tester of query complexity $q(\epsilon)$.*
2. *There exists a function $T : (0,1] \rightarrow \mathbb{N}$ such that for every $\epsilon > 0$, the property Π is ϵ-approximated by the union of $T(\epsilon)$ regularity properties of fineness $1/T(\epsilon)$.*

5 A Taxonomy of the known results

Teaching note: The current section is a kind of digest of the material presented in Sections 2–4, organized according to the query complexity of the various property testing problems. In addition it presents two results: A query complexity hierarchy (Theorem 24) and a result asserting that the non-adaptive testers can achieve query complexity that is at most quadratic in the query complexity of an arbitrary tester (Theorem 25). Actually, the tester derived in Theorem 25 is even more restricted: It merely inspects the subgraph induced by a random sample of vertices.

Testers of query complexity that depends on the size of the graph. We first mention that graph properties of arbitrary query complexity are known: Specifically, in this model, graph properties (even those in $\mathcal{P}$) may have query complexity ranging from $O(1/\epsilon)$ to $\Omega(k^2)$, where k denotes the number of vertices, and the same holds also for *monotone graph properties*[52] in $\mathcal{NP}$. One of these hierarchy theorems states (cf. [27]).

Theorem 24 (query hierarchy for testing graph properties in the dense graph model): *For every $q : \mathbb{N} \rightarrow \mathbb{N}$ that is at most quadratic such that $k \mapsto \lfloor \sqrt{q(k)} \rfloor$ is onto, there exists a graph property Π and $\epsilon > 0$ such that ϵ-testing Π on k-vertex graphs has query complexity $\Theta(q(k))$. Furthermore, if $k \mapsto q(k)$ is computable in $\text{poly}(\log k)$-time, then Π is in $\mathcal{P}$, and the tester is relatively efficient in the sense that its running time is polynomial in the total length of its queries.*

[52] A graph property Π is called monotone if, for every $G \in \Pi$, the graph obtained from G by adding any edge to G is also in Π. The same result holds for anti-monotone properties (where omitting edges preserves the property).

We mention that the testers used in the upper bound have query complexity $\text{poly}(1/\epsilon) \cdot q(k)$.

Theorem 24 is established in [27] by using unnatural graph properties, starting from the $\Omega(k^2)$ lower bound of [26], which also uses an unnatural graph property.[53] In contrast, the $\Omega(k)$ lower bound established in [19] (following [2]) refers to the natural property of testing whether an k-vertex graph consists of two isomorphic copies of some $k/2$-vertex graph.

Testers of query complexity that is independent of the size of the graph. In this section, we focus on properties that can be tested within *query complexity that only depends on the proximity parameter* (i.e., ϵ); that is, *the query complexity does not depend on the size of the graph being tested.* As we have seen, there is much to say about this class of properties. For $q : (0,1] \to \mathbb{N}$, let $\mathcal{C}(q)$ denote the class of graph properties that can be tested within query complexity q. We shall focus on three classes of properties.

1. *Arbitrary q such that $q(\epsilon) \gg \text{poly}(1/\epsilon)$.* By Corollary 18, triangle-freeness is in the class $\mathcal{C}(q)$, for some (tower-of-exponents) function q, but is not in the class $\mathcal{C}(q)$ for any polynomial q. The same holds for H-freeness for any non-bipartite H.

2. *The case of $q(\epsilon) = \text{poly}(1/\epsilon)$.* By Theorem 12, every graph partition problem is in the class $\mathcal{C}(q)$, for some polynomial q. In particular, t-`Colorability` is in $\mathcal{C}(q_t)$ where $q_2(\epsilon) = \widetilde{O}(\epsilon^{-2})$ and $q_k(\epsilon) = \widetilde{O}(\epsilon^{-4})$ for any $k \geq 3$ (see Theorem 13). It is also known that $q_2(\epsilon)$ cannot be $o(\epsilon^{-3/2})$.

   By Theorem 5, degree regularity is in $\mathcal{C}(q)$ for $q(\epsilon) = O(1/\epsilon^2)$, and $q(\epsilon)$ cannot be $o(1/\epsilon^2)$ (see Exercise 1).

3. *The case of $q(\epsilon) = \widetilde{O}(1/\epsilon)$.* By Proposition 6, Biclique is in $\mathcal{C}(q)$ for $q(\epsilon) = O(1/\epsilon)$. As mentioned in Footnote 40, the same bound holds for H-freeness for any bipartite H. Additional properties in this class are reviewed in Section 5.3.

Before further discussing the foregoing results, we mention that, when disregarding a possible quadratic blow-up in the query complexity, we may assume that the tester is non-adaptive. Furthermore, it is actually canonical in the following sense.

Theorem 25 (canonical testers [34, Thm 2]):[54] *Let Π be any graph property. If there exists a tester with query complexity q for Π, then there exists a tester for Π that uniformly selects at random a set of $O(q)$ vertices and accepts if and only if the induced subgraph has property Π', where Π' is a graph property that may depend on the number of vertices in the tested graph* (i.e., k) *as well as on Π. Furthermore, if the original tester has one-sided error, then so does the new tester, and a sample of $2q$ vertices suffices*

Indeed, the resulting tester is called canonical. In particular, it *decided based on an inspection of the subgraph induced by a random sample of vertices* (and, thus, is, in particular, non-adaptive). We

[53] This is a common phenomenon in hierarchy theorems; cf. [23, Chap. 4].

[54] As pointed out in [9], the statement of [34, Thm 2] should be corrected such that the auxiliary property Π' may depend on k and not only on Π. Thus, on input k and ϵ (and oracle access to an k-vertex graph G), the canonical tester checks whether a random induced subgraph of size $s = O(q(k,\epsilon))$ has the property Π', where Π' itself (or rather its intersection with the set of s-vertex graphs) may depend on k. In other words, the tester's decision depends only on the induced subgraph that it sees and on the size parameter k.

warn that Π' need not equal Π (let alone that Π' may depend on k). Still, in many natural cases, $\Pi' = \Pi$ (e.g., t-`Colorability`). We warn that, in addition to the (possible) quadratic blow-up in the query complexity, the time complexity of the canonical tester may be significantly larger than the time complexity of the original tester.

5.1 Testability in $q(\epsilon)$ queries, for any function q

Recall that Theorem 18 (Part 1) implies that all subgraph freeness properties have constant-query (one-sided error) proximity-oblivious testers. Also, Theorem 23 provides a combinatorial characterization of the class of properties that can be tested within query complexity that only depends on the proximity parameter.

The downside of the algorithms that emerge from the aforementioned results is that their query complexity is related to the proximity parameter via a function that grows tremendously fast. Specifically, in the general case, the query complexity is only upper bounded by a tower of a tower of exponents (in a monotonically growing function of $1/\epsilon$, which in turn depends on the property at hand). Furthermore, it is known that a super-polynomial dependence on the proximity parameter is inherent to the foregoing result. Actually, as shown by Alon [1], such a dependence is essential even for testing *triangle freeness*.

The latter fact provides a nice demonstration of the non-triviality of testing graph properties. *One might have guessed that $O(1/\epsilon)$ or $O(1/\epsilon^3)$ queries would have sufficed to detect a triangle in any graph that is ϵ-far from being triangle free, but Alon's result asserts that this guess is wrong and that* $\text{poly}(1/\epsilon)$ *queries do not suffice.* We mention that the best upper bound known for the query complexity of testing triangle freeness is $\mathtt{T}(\text{poly}(1/\epsilon))$, where $\mathtt{T}$ is the tower function defined inductively by $\mathtt{T}(n) = \exp(\mathtt{T}(n-1))$ with $\mathtt{T}(1) = 2$ (cf. [1]).

Perspective: Is it all about combinatorics? Theorem 25 seems to suggest that the study of testing graph properties (in this model) reduces to combinatorics, since it asserts that *testing reduces to inspecting a random induced subgraph* (of the corresponding size). This lesson is made more concrete by the characterization of "size-oblivious" testable graph properties provided by Theorem 23, which refers to the notion of a *regularity property*, where regularity is in the sense of Szeme ́redi's Regularity Lemma [46]. Recall that this result essentially asserts that a graph property can be tested in query complexity that only depends on ϵ if and only if it can be characterized in terms of a constant number of regularity properties. In retrospect, this justifies the use of the Regularity Lemma in the proof of (Part 1 of) Theorem 18. In any case, the lesson is that, when ignoring the specific dependency on ϵ, *testing graph properties in query complexity that only depends on ϵ reduces to testing the edge densities of pairs in a regular partition.* However, as noted already and further advocated next, *this lesson ignores both the running time of the tester and the exact value of the query complexity.*

Perspective: The exact query complexity does matter. It is indeed an amazing fact that many properties can be tested within (query) complexity that only depends on the proximity parameter (rather than also on the size of the object being tested). This amazing statement seems to put in shadow the question of the form of the aforementioned dependence, and blurs the difference between a reasonable dependence (e.g., a polynomial relation) and a prohibiting one (e.g., a tower-function relation). We beg to disagree with this sentiment and claim that, as in the context of standard approximation problems (cf. [38]), *the dependence of the complexity on the approximation* (or proximity) *parameter is a key issue.*

We wish to stress that we do value the impressive results of [2, 5, 6, 7, 21] (let alone [3]), which refer to graph property testers having query complexity that is independent of the graph size but depends prohibitively on the proximity parameter. We view such results as an impressive first step, which called for further investigation directed at determining the actual dependency of the complexity on the proximity parameter.

While it is conceivable that there exist (natural) graph properties that can be tested in $\exp(1/\epsilon)$ queries but not in $\mathrm{poly}(1/\epsilon)$ queries, we are not aware of such a property. (Needless to say, demonstrating the existence of such (natural) properties is an interesting open problem.) We thus move directly from complexities of the form $\mathtt{T}(1/\epsilon)$ (and larger) to complexities of the form $\mathrm{poly}(1/\epsilon)$.

5.2 Testability in $\mathrm{poly}(1/\epsilon)$ queries

Testers of query complexity $\mathrm{poly}(1/\epsilon)$ are known for several natural graph properties, which fall under the general framework of *graph partition problems* (presented and studied in Section 3). We briefly recall some of these properties, while reminding the reader that by Theorem 12, every graph partition problem is testable in $\mathrm{poly}(1/\epsilon)$ queries.

- t-`Colorability`, for any fixed $t \geq 2$.

  Recall that by Theorem 13, t-`Colorability` has a one-sided error tester of query complexity $\widetilde{O}(t^2/\epsilon^4)$ for any $t > 2$. For $t = 2$ this tester has query-complexity (and running-time) $\widetilde{O}(1/\epsilon^2)$.

- ρ-`Clique`, for any fixed $\rho > 0$, where ρ-`Clique` is the set of graphs that have a clique of density ρ (i.e., k-vertex graphs having a clique of size ρk).

- ρ-`Cut`, for any fixed $\rho > 0$, where ρ-`Cut` is the set of graphs that have a cut of density at least ρ (compared to k^2).

- ρ-`Bisection`, for any fixed $\rho > 0$, where ρ-`Bisection` is the set of graphs that have a bisection of density at most ρ (i.e., an k-vertex graph is in ρ-Bisection if its vertex set can be partitioned into two equal parts with at most ρk^2 edges going between them).

Except for k-`Colorability`, all the other testers have two-sided error, and this is unavoidable for any tester of $o(k)$ query complexity for any of these properties.

Beyond graph partition problems. Although many natural graph properties can be formulated as partition problems, many other properties that can be tested with $\mathrm{poly}(1/\epsilon)$ queries cannot be formulated as such problems. The list include the set of regular graphs, connected graphs, planar graphs, and more. We identify three classes of such natural properties:

1. Properties that only depends on the vertex degree distribution (e.g., degree regularity and average degree). For example, for any fixed $\rho > 0$, the set of k-vertex graphs having ρk^2 edges can be tested using $O(1/\epsilon^2)$ queries, which is the best result possible.[55] The same holds with respect to testing degree regularity (see Theorem 5 and Exercise 1).

[55] Both the upper and lower bounds can be proved by reduction to the problem of estimating the average value of Boolean functions (cf. [18]).

2. Properties that are satisfied only by sparse graphs (e.g., k-vertex graphs having $O(k)$ edges) such as `Cycle-freeness` and `Planarity`. See Proposition 4 for a more general statement.

3. Properties that are almost trivial in the sense that, for some constant $c > 0$ and every $\epsilon > k^{-c}$, all k-vertex graphs are ϵ-close to the property (see Proposition 3). For example, every k-vertex graph is k^{-1}-close to being connected (or being Hamiltonian or Eulerian).

In view of all of the foregoing, we believe that characterizing the class of graph properties that can be tested in poly$(1/\epsilon)$ queries may be very challenging. We mention that the special case of induced subgraph freeness properties was resolved in [8].

5.3 Testability in $\widetilde{O}(1/\epsilon)$ queries

While Theorem 25 may be interpreted as suggesting that testing in the dense graph model leaves no room for algorithmic design, this conclusion is valid only if one ignores a possible quadratic blow-up in the query complexity (and also disregards the time complexity). As advocated in [30], a finer examination of the model, which takes into account the exact query complexity (i.e., cares about a quadratic blow-up), reveals the role of algorithmic design. In particular, the results in [30] distinguish adaptive testers from non-adaptive ones, and distinguish the latter from canonical testers. These results refer to testability in $\widetilde{O}(1/\epsilon)$ queries. In particular, it is known that:[56]

- Testing every "non-trivial for testing" graph property requires $\Omega(1/\epsilon)$ queries, even when adaptive testers are allowed. Furthermore, any canonical tester for such a property requires $\Omega(1/\epsilon^2)$ queries, since it must inspect a subgraph that is induced by $\Omega(1/\epsilon)$ vertices.

- There exist an infinite class of natural graph properties that can be tested by $\widetilde{O}(1/\epsilon)$ non-adaptive queries. Specifically, this class contains all properties obtained by an (uneven) blow-up of some fixed graph.[57]

- There exists a natural graph property that can be tested by $\widetilde{O}(1/\epsilon)$ adaptive queries, requires $\Omega(\epsilon^{-4/3})$ non-adaptive queries, and is actually testable by $O(\epsilon^{-4/3})$ non-adaptive queries. The property for which this is shown is called `Clique Collection`, and contains all graphs that consist of a collection of isolated cliques. That is, *the problem of testing* `Clique Collection` *has* (general) *query complexity* $\widetilde{\Theta}(1/\epsilon)$ *and non-adaptive query complexity* $\Theta(\epsilon^{-4/3})$.

- There exists a natural graph property that can be tested by $\widetilde{O}(1/\epsilon)$ adaptive queries but requires $\Omega(\epsilon^{-3/2})$ non-adaptive queries. The property for which this is shown is called `Biclique Collection`, and contains all graphs that consist of a collection of isolated bicliques.

All the above testers have one-sided error probability and are efficient, whereas the lower bounds hold also for two-sided error testers (regardless of efficiency).

The foregoing results seem to indicate that even at this low complexity level (i.e., testing in $\widetilde{O}(1/\epsilon)$ adaptive queries) there is a lot of structure and much to be understood. In particular, it is

[56] With the exception of the result regarding testability by $\widetilde{O}(1/\epsilon)$ non-adaptive queries, all other results are due to [30]. The exceptional result was prove in a subsequent work of [11], which extended a corresponding result of [30], which in turn referred to the special case in which H is a t-clique.

[57] That is, for any fixed graph $H = ([t], F)$, a k-vertex blow-up of H is a k-vertex graph obtained by replacing each vertex of H by an independent set (of arbitrary size), called a cloud, and connecting the vertices of the i^{th} and j^{th} clouds by a biclique if and only if $\{i, j\} \in F$.

conjectured in [30] that, *for every $t \geq 4$, there exist graph properties that can be tested by $\widetilde{O}(1/\epsilon)$ adaptive queries and have non-adaptive query complexity* $\Theta(\epsilon^{-2+\frac{2}{t}})$. Partial progress towards establishing this conjecture is presented in [30].

5.4 Additional issues

Let us highlight some issues that arise from the foregoing exposition.

Adaptive testers versus non-adaptive ones. Recall that Theorem 25 asserts that canonical testers (which are, in particular, non-adaptive) have query complexity that is at most quadratic in the query complexity of general (possibly adaptive) testers. The results surveyed in Section 5.3 indicate that a polynomial gap may exist: There is a (natural) property that can be ϵ-tested by $\widetilde{O}(1/\epsilon)$ adaptive queries, but requires $\Omega(1/\epsilon^{3/2})$ non-adaptive queries. Furthermore, it was conjectured that *for every $c < 2$, there exist graph properties that can be tested by $\widetilde{O}(1/\epsilon)$ adaptive queries and has non-adaptive query complexity* $\Theta((1/\epsilon)^c)$. Here we propose a possibly easier goal:

Open Problem 26 (a maximal gap between adaptive and non-adaptive queries): *Show that, for every $c < 2$, there exist graph properties that can be tested by $q(\epsilon) = \Omega(1/\epsilon)$ adaptive queries but requires $\Theta(q(\epsilon)^c)$ non-adaptive queries.*

A different question, raised by Michael Krivelevich, is whether (adaptive versus non-adaptive complexity) gaps exists also for properties having query complexity that is significantly larger than $\widetilde{O}(1/\epsilon)$; that is, does there exist a graph property that, for some $c > 1$, has adaptive query complexity $q(\epsilon) \geq (1/\epsilon)^c$ and non-adaptive query complexity $\Omega(q(\epsilon)^c)$? Recall that ϵ-testing `Bipartiteness`, which has non-adaptive query complexity $\widetilde{\Theta}(\epsilon^{-2})$ (cf. [4, 15])[58] and requires $\Omega(\epsilon^{-3/2})$ adaptive queries [15], may be testable in $\epsilon^{-(2-\Omega(1))}$ adaptive queries (cf. [14]).

One-sided versus two-sided error probability. As noted above, for many natural properties there is a significant gap between the complexity of one-sided and two-sided error testers. For example, ρ-`Cut` has a two-sided error tester of query complexity poly$(1/\epsilon)$, but no one-sided error tester of query complexity $o(k^2)$ where k is the number of vertices in the tested graph. In general, the interested reader may contrast the characterization of two-sided error testers in [3] with the results in [7].

Proximity Oblivious Testers. Some of the positive results regarding property testing were obtained by presenting (one-sided error) proximity oblivious testers (of constant-query complexity and detection probability that depends only on the distance of the tested graph from the property). Furthermore, Theorem 20 provided a simple characterization of properties having such testers. It follows that constant-query proximity-oblivious testers do not exist for many easily testable properties (e.g., `Bipartiteness` (see Exercise 4)). Furthermore, even when proximity-oblivious testers exist, repeating them does not necessarily yield the best standard testers for the corresponding property (see, e.g., `Clique Collection` [31]).

[58] The $\widetilde{O}(\epsilon^{-2})$ upper bound is due to [4], improving over [26], whereas the $\Omega(\epsilon^{-2})$ lower bound is due to [15].

Tolerant testing. Recall that property testing calls for distinguishing objects having a predetermined property from object that are far from any objects that has this property (i.e., are far from the property). A more "tolerant" notion requires distinguishing objects that are close to having the property from objects that are far from this property. Such a distinguisher is called a tolerant tester, and is a special case of a distance approximator that given any object is required to approximate its distance to the property. The general study of these related notions (which are applicable to all three models discussed in Section 1) was initiated by Parnas, Ron, and Rubinfeld [43].

A simple observation is that any tester that makes uniformly distributed queries offers some level of tolerance. Specifically, if a tester makes $q(\epsilon)$ queries and each query is uniformly distributed, then this tester distinguishes between objects that are ϵ-far from the property and objects that are $(\epsilon/10q(\epsilon))$-close to the property. Needless to say, the challenge is to provide stronger relations between property testing and distance approximators. Such a result was provided by Fischer and Newman [21]: They showed that *any graph property that can be tested in a number of queries that only depends on the proximity parameter, has a distance approximator of query complexity that only depends on the proximity parameter.*[59]

Directed graphs. Our discussion was confined to undirected graphs. Nevertheless, the three models discussed in Section 1 extend naturally to the case of directed graphs. In particular, in the dense graph model, a directed graph is represented by its adjacency matrix, which is possibly asymmetric; that is, the $(i,j)^{\text{th}}$ entry in the matrix is 1 if and only if there is a directed edge from the i^{th} vertex to the j^{th} vertex. The study of testing properties of directed graphs was initiated by Bender and Ron [13]. In particular, in the dense graph model, they showed a poly$(1/\epsilon)$-query tester for `Acyclicity` (i.e., the set of directed graphs that contain no directed cycles). Testing directed graphs in the dense graph model was further studied in [5], which focuses on testing subgraph-freeness.

6 Final notes

It should not come as a surprise that this relatively long lecture notes have a relatively long section of final notes. Following the usual historical notes and before the usual exercises, we insert a discussion of property testing versus other forms of approximation (Section 6.2) as well as other reflections (Section 6.3).

6.1 Historical perspective and credits

The study of property testing in the dense graph model was initiated by Goldreich, Goldwasser, and Ron [26], as a concrete and yet general framework for the study of property testing at large. From that perspective, it was most natural to represent graphs as Boolean functions, and the adjacency matrix representation was the obvious choice. This dictated the choice of the type of queries as well as the distance measure, leading to the definition of the dense graph model.

Testing graph properties in the dense graph model has attracted a lot of attention. Among the directions explored are the study of the complexity of specific natural properties [26, 4, 15, 35, 19], attempts to explore general classes of easily testable properties [26, 2, 1], and characterizations of

[59] This result is implied by Theorem 23, but it was proved in [21] before the latter theorem was proved in [3]. In fact, the ideas in [21] paved the road to [3].

classes of properties that are testable under various restrictions (e.g., [1, 5, 34, 6, 7, 8, 21, 3, 17]). In addition, many studies of property testing at large have devoted special attention to testing graph properties in the dense graph model [27, 31, 33, 32]. Some of the aforementioned works as well as some that were not listed will be further discussed below.

Before proceeding, we comment on the relation between the dense graph model and the other two models that were briefly presented in Section 1 and will be the topic of the two subsequent lectures. In retrospect, the dense graph model seems most natural when graphs are viewed as representing generic (symmetric) binary relations. But, in many other setting, the other two models are more natural. Needless to say, the general graph model is the most general one, and it is indeed closest to actual algorithmic applications. In other words, this model is relevant for most applications, since these seem to refer to general graphs (which model various natural and artificial objects). In contrast, the dense graph model is relevant to applications that refer to (dense) binary relations over finite sets, whereas the bounded-degree graph model is relevant only to applications in which the vertex degree is bounded. The study of testing graph properties in the bounded-degree graph model was initiated by Goldreich and Ron [29], whereas the study of the general model was initiated by Parnas and Ron [42] and generalized to its current form by Kaufman, Krivelevich, and Ron [39].[60]

Simple properties: trivial, sparse, and degree-regularity. The results presented in Sections 2.2 and 2.3 are taken from [26], with the exception of the improved bound stated in Theorem 5. The latter improvement (over [26, Prop. 10.2.1.3]) appeared in [24, Apdx A.1], but the proof of Claim 5.1 is reproduced from [33, Apdx A.1].

The strategy underlying Algorithm 5.2 can be traced to the last paragraph of Levin's work on one-way functions and pseudorandom generators [40, Sec. 9], and is stated explicitly in [28, Lem. 3] (see [22, Clm. 2.5.4.1] for an alternative presentation). Within the context of property testing, this strategy was first used in [29] (see Lemma 3.3 in the proceeding version and Lemma 3.6 in the journal version).

Testing general partition problems. The framework of general graph partition problems was introduced by Goldreich, Goldwasser, and Ron [26], and the testers for all properties in it (as summarized by Theorem 12) constitute the main results in their paper. We chose to present only the analysis of the Bipartiteness tester (i.e., Lemma 8, which is taken from [26]). The improved testers for t-`Colorability` (captured by Theorem 13) are due to Alon and Krivelevich [4].

Using Szemeŕedi's Regularity Lemma. In retrospect, it turns out that testers for k-`Colorability` were implicit in works of Bollobas *et al.* [16] and Rodl and Duke [44], referring to $k = 2$ and $k > 2$, respectively. These works, which predate the definition of property testing, use the regularity lemma, and obtain testers of correspondingly huge query complexity (i.e., a tower of poly$(1/\epsilon)$ exponents). Testers for subgraph freeness which are also based on the regularity lemma, were presented by Alon *et al.* [2]; the corresponding result is stated in Part 1 of Theorem 18. Several subsequent works also used the regularity lemma (or new extensions of it), culminating with the work of Alon *et al.* [3], to be reviewed next.

[60] Parnas and Ron [42] only allowed incidence queries (like in the bounded-degree graph model), and Kaufman, Krivelevich, and Ron [39] also allowed adjacency queries (as in the dense graph model).

Characterizations. The celebrated result of Alon, Fischer, Newman, and Shapira [3] provides a combinatorial characterization of the class of properties that can be tested within query complexity that only depends on the proximity parameter (see Theorem 23). We view the result more as a structural result regarding properties that can be tested within such a complexity (than as a characterization). It asserts that these properties can be approximated by finite unions of "regularity properties" (where each regular property is a set of graphs that has a regular partition with certain edge densities).[61] A result of a similar flavour was proved independently by Borgs *et al.* [17], while referring to "graph limits".

The class of graph properties that can be tested within query complexity that only depends on the proximity parameter ϵ, contains natural properties that are not testable in query complexity $\text{poly}(1/\epsilon)$; see [1]. A begging open problem is to characterize the class of graph properties that are testable in $\text{poly}(1/\epsilon)$ queries.

Open Problem 27 (characterization of graph properties that are testable in $\text{poly}(1/\epsilon)$ queries): *Characterize the class of graph properties that can be tested, in the dense graph model, within query complexity that is polynomial in the reciprocal of the proximity parameter.*

This problem has been resolved for the class of subgraph freeness properties [5] (see Theorem 18). It will be interesting to find other classes of natural graph properties that are "split" among those having $\text{poly}(1/\epsilon)$-query testers and those having $F(1/\epsilon)$-query testers only for some super-polynomial function F.

The characterization of graph properties that have constant-query (one-sided error) proximity oblivious testers (i.e., Theorem 20) is due to Goldreich and Ron [31], which build on [8] for constructing testers and on [34] for inferring that such testers exist only for induced subgraph freeness properties.

Canonical testers and the power of adaptivity. The notion of canonical testers and Theorem 25 are due to Goldreich and Trevisan [34]. Theorem 25 explains that the fact that almost all prior testers, in the dense graph model, work by inspecting a random induced subgraph is no coincidence, since the query complexity of such testers is at most quadratic in the query complexity of the best possible tester. Complexity gaps between canonical testers and general non-adaptive testers, and between the latter and general adaptive testers were shown by Goldreich and Ron [30]. While the demonstrated gap for the first case it optimal (i.e., it matches the quadratic upper bound), the gap shown in the second case is not optimal (see Problem 26).

6.2 Testing versus other forms of approximation

We shortly discuss the relation of the notion of approximation underlying the definition of testing graph properties (in the dense graph model)[62] to more traditional notions of approximation. Throughout this section, we refer to randomized algorithms that have a small error probability, which we ignore for simplicity.

[61] These are regular partitions in the sense of Szemeredi's Regularity Lemma [46], and the specified edge densities may be different for each regular pair.

[62] Analogous relations hold also in the other models of testing graph properties.

Application to the standard notion of approximation. The relation of testing graph properties to standard notions of approximation is best illustrated in the case of `Max-CUT`. Any tester for ρ-`Cut`, working in time $T(\epsilon, k)$, yields an algorithm for approximating the size of the maximum cut in an k-vertex graph, up to additive error ϵk^2, in time $\widetilde{O}(\log(1/\epsilon)) \cdot T(\epsilon, k)$.[63] Thus, for any constant $\epsilon > 0$, using the tester of Theorem 12, we can approximate the size of the max-cut to within ϵk^2 in constant time. This yields a *constant-time approximation scheme* (i.e., to within any constant relative error) for dense graphs. Finding an approximate max-cut does not seem to follow from the mere existence of a tester for ρ-cut; yet, the tester of Theorem 12 can be used to find such a cut in time linear in k.

Relation to "dual approximation" (cf. [38, Chap. 3]). To illustrate this relation, we consider the tester for ρ-`Clique`. The traditional notion of approximating `Max-Clique` corresponds to distinguishing the case in which the given k-vertex graph has a clique of size ρk from, say, the case in which the graph has no clique of size $\rho k/2$. On the other hand, when we talk of testing ρ-`Clique`, the task is to distinguish the case in which an k-vertex graph has a clique of size ρk from the case in which it is ϵ-far from the class of k-vertex graphs having a clique of size ρk. This is equivalent to the "dual approximation" task of distinguishing the case in which an k-vertex graph has a clique of size ρk from the case in which any ρk-subset of the vertices misses at least ϵk^2 edges. To demonstrate that these two tasks are vastly different, we mention that whereas the former task is NP-Hard for $\rho < 1/4$ (see [12, 37]), the latter task can be solved in $\exp(O(1/\epsilon^2))$-time, for any $\rho, \epsilon > 0$. We believe that there is no absolute sense in which one of these approximation tasks is more important than the other: Each of these tasks may be relevant in some applications and irrelevant in others.

6.3 Two additional points

Let us reflect about some issues that arise from the foregoing exposition.

A contrast to recognizing graph properties. The notion of testing a graph property Π is a *relaxation* of the classical notion of *recognizing the graph property* Π, which has received much attention since the early 1970's (cf. [41]). In the classical (recognition) problem there are no margins of error; that is, one is required to accept all graphs having property Π and reject all graphs that lack property Π. In 1975, Rivest and Vuillemin resolved the Aanderaa–Rosenberg Conjecture, showing that any deterministic procedure for deciding any non-trivial monotone k-vertex graph property must examine $\Omega(k^2)$ entries in the adjacency matrix representing the graph. The query complexity of randomized decision procedures was conjectured by Yao to be $\Omega(k^2)$, and the currently best lower bound is $\Omega(k^{4/3})$. This stands in striking contrast to the aforementioned results regarding testing graph properties that establish that many natural (non-trivial) monotone graph properties can be *tested* by examining a constant number of locations in the matrix (where this constant depends on the constant value of the proximity parameter).

[63] Note that is a graph G is ϵ-close to having a ρ-cut, then it must have a cut of size at least $(\rho - 0.5\epsilon) \cdot k^2$. (This is since G' is ϵ-close to a graph G' that has a ρ-cut, and this very cut only misses $\epsilon k^2/2$ edges in G.) Hence, if the tester accepts G with probability at least 2/3, then G must have a $(\rho - 0.5\epsilon)$-cut. The $\widetilde{O}(\log(1/\epsilon)$ factor accounts for a binary search (for the highest value of $\rho \in \{\epsilon, 2\epsilon, ..., \lfloor 1/\epsilon \rfloor \cdot \epsilon\}$) as well as for error reduction needed for invoking the tester $\log(1/\epsilon)$ times.

Graph properties are poor codes. We note that with the exception of two properties, which each contains a single k-vertex graph, the adjacency matrix representation of any property Π_k of k-vertex graphs yields a code over $\{0,1\}^{\binom{k}{2}}$ with relative distance at most $O(1/k)$. Specifically, if Π_k neither consists of the k-vertex clique nor of the k-vertex independent set, then Π_k contains a graph $G = ([k], E)$ that contains two vertices $i, j \in [k]$ that have different neighborhoods in G. Consider a permutation π that transposes i and j, while leaving the rest of $[k]$ intact, and let $G' = ([k], \{\{\pi(u), \pi(v)\} : (u,v) \in E\})$.[64] Then $G' \in \Pi_k$, but G' is $\frac{2k}{k^2}$-close to G.

6.4 Exercises

The exercises in this section seem more interesting than the ones in prior lectures.

Exercise 1 (query complexity lower bound for testing degree regularity): *Prove that ϵ-testing degree regularity requires $\Omega(1/\epsilon^2)$ queries.*

Guideline: Show that distinguishing the following two sets of graphs requires $\Omega(1/\epsilon^2)$ queries. The first set consists of k-vertex graphs that consist of two equal-sized connected components such that each component is $0.25k$-regular. The second set is similar except that one connected components is $(0.25+\epsilon)\cdot k$-regular and the other is $(0.25-\epsilon)\cdot k$-regular. Reduce from the problem of estimating the average of a Boolean function defined on a large set (see [18]). Specifically, first reduce the problem of distinguishing functions $f : [k] \rightarrow \{0,1\}$ that have average value 0.5 from functions $f : [k] \rightarrow \{0,1\}$ that have average value $0.5 + \epsilon$ to the problem of distinguishing pairs of functions $f_1, f_2 : [k] \rightarrow \{0,1\}$ that have equal average value (of 0.5) from pairs of functions that have an average that differs by at least 2ϵ.[65] Next, reduce the latter problem to the one about graphs.[66]

Exercise 2 (On Levin's economical work investment strategy): *In continuation to Section 2.4, show that the goal can be achieved by investing $O(1/\epsilon^c)$ work if $c > 1$ and the work invested in element ω is $\widetilde{O}(1/q(\omega)^c)$. Also show that if the work invested in ω is $O(1/q(\omega))$, then the goal can be achieved by investing $(\epsilon^{-1}\log(1/\epsilon))$ work.*

Guideline: Suppose that the work invested in ω is $((\log(1/q(\omega))^d/q(\omega)^c)$. Then, for $c > 1$, selecting $O(i^{d+2} \cdot 2^i)$ points (for each $i \in [\ell]$), and investing $O(1/2^i\epsilon)^c$ work in each of them, will do. For $c = 1$, selecting $O(\log(1/\epsilon))^{d+1} \cdot 2^i$ points (for each $i \in [\ell]$), yields a better result.

Exercise 3 (testing d-regularity): *For any fixed $\rho > 0$, prove that ϵ-testing if a k-vertex graph is $\lfloor \rho k \rfloor$-regular can be done by $O(1/\epsilon^2)$ non-adaptive queries.*

Guideline: Use an adaptation of the proof of Theorem 5.

[64] That is, the adjacency matrix representing G' is obtained from the adjacency matrix representing G by switching the i^{th} and j^{th} rows (and ditto for the columns).

[65] For example, map f to the pair $(f, f \oplus 1)$.

[66] First replace $[k]$ by $\mathbb{Z}_k$. Then, for each $\sigma \in \{1,2\}$ and $i, j \in \mathbb{Z}_k$, let $g_\sigma(i,j) = f_\sigma(i + j \bmod k)$ if $i \neq j$ and $g_\sigma(i,i) = 0$ otherwise. Finally, consider the graph represented by the adjacency predicate $g : \mathbb{Z}_{2k}^2 \rightarrow \{0,1\}$ such that for every $i, j \in \mathbb{Z}_k$ it holds that $g(i,j) = g_1(i,j)$ and $g(i+k, j+k) = g_2(i,j)$, where $g(i,j) = 0$ if $i \in \mathbb{Z}_k$ and $j \in \mathbb{Z}_{2k} \setminus \mathbb{Z}_k$ (or vice versa).

Exercise 4 (Bipartiteness has no proximity oblivious tester):[67] *Prove that* Bipartiteness *has no proximity oblivious tester that makes a constant number of queries. Start with the case of one-sided error.*

Guideline: The following two notions are useful towards a solution. The odd-girth of a graph is the length of the shortest odd cycle in it.[68] A m-factor blow-up of a graph $H = ([\ell], F)$ is a $m \cdot \ell$-vertex graph obtained by replacing each vertex of H by an independent set of size m, called a cloud, and connecting the vertices of the i^{th} and j^{th} clouds by a biclique if and only if $\{i, j\} \in F$. The one-sided error case can be handled by considering, for every q, an arbitrary k-vertex graph that has odd-girth greater than q and is $\Omega(1/q^2)$-far from being bipartite. (For example, consider a k/ℓ-factor blow-up of an ℓ-cycle, where $\ell = 2\lceil q/2 \rceil + 1$.)[69] Then, a q-query proximity oblivious tester (POT) must reject this graph with positive probability, although it saw no cycle in it, which means that this POT cannot have one-sided error.

Moving to the two sided-error case, suppose towards the contradiction that a (two-sided error) POT that makes q queries exists. Let $\ell = 2\lceil q/2 \rceil + 1$, and consider the following two distributions on k-vertex graphs (for each k that is a multiple of $2 \cdot \ell$):

1. Uniform distribution over all graphs that are obtained by a $k/2\ell$-factor blow-up of a 2ℓ-cycle.
2. Uniform distribution over all graphs that are obtained by a $k/2\ell$-factor blow-up of a graph that consists of two disjoint ℓ-cycles.

Note that the graphs in the first distribution are bipartite, whereas all graphs in the second distribution are $\Theta(1/q^2)$-far from being bipartite. The key observation is that these two distributions are perfectly indistinguishable by a machine that makes q queries. This claim is proved by showing that the answers provided by these two distributions on any sequence of queries is identically distributed (see exercise in the notes on lower bound techniques).[70]

Exercise 5 (testers for Bipartiteness must inspect $\Omega(1/\epsilon)$ vertices):[71] Bipartiteness *can not be ϵ-tested by an algorithm whose queries touch $o(1/\epsilon)$ vertices.* (Equivalently, if an ϵ-tester for Bipartiteness inspects the subgraph induced by $s(\epsilon)$ vertices, then $s(\epsilon) = \Omega(1/\epsilon)$.)

Guideline: Consider the following two distributions on k-vertex graphs. In both distributions, one selects uniformly a 3-partition (V_0, V_1, V_2) such that $|V_0| = 3\epsilon k$ and $|V_1| = |V_2| = (1-3\epsilon)k/2$. In the first distribution bicliques are placed between each pair of parts, whereas in the second distribution

[67] Based on a result in [31].

[68] For general perspective, we mention that the girth of a graph is the length of the shortest cycle in it, and that a k-vertex graph of girth g can have at most $k^{1+\Theta(1/g)}$ edges.

[69] To see that this graph is $\Omega(1/q^2)$-far from being bipartite, consider the omission of any set of $0.1 \cdot m^2$ edges, where $m = k/\ell$. Call a vertex good if it has at least $\frac{5}{3} \cdot m$ edges (in the resulting graph), and note that at least 0.6 of the vertices in each cloud are good. Now, pick a good vertex v_1 in the first cloud, and for $i = 2, ..., \ell - 1$ pick a good vertex v_i in the i^{th} cloud such that v_i is adjacent to v_{i-1}. (Such a choice exists, since v_{i-1} has at least $2m/3$ neighbors in the $i^x th$ cloud and at most $0.4m$ of them are not good.) Observing that both v_1 and $v_{\ell-1}$ have each $2m/3$ neighbors in the ℓ^{th} cloud, the claim follows. (We mention that the same argument establishes the existence of at least $(0.1m)^\ell$ different ℓ-cycles in the resulting graph.)

[70] It is instructive to think that each pair query (u, v) is answered by either 0 (indicating that $\{u, v\}$ is not an edge) or by $\sigma \in \{\pm 1\}$, where $\sigma = 1$ (resp., $\sigma = -1$) indicates that v resides in the cloud succeeding (resp., preceding) the cloud in which u resides.

[71] Based on a result in [4].

a biclique is placed only between V_1 and V_2. Then, each graph in the first distribution is ϵ-far from being bipartite (because there are $3\epsilon k \cdot ((1-3\epsilon)k/2)^2$ triangles, whereas each edge participates in less than $k/2$ triangles). Yet, an algorithm that "inspects" $o(1/\epsilon)$ vertices is unlikely to distinguish the two distributions (since it is unlikely to inspect any vertex of V_0).

Exercise 6 (a random induced subgraph preserves the distance from being bipartite): *Prove that if $G = ([k], E)$ is ϵ-far from being bipartite, then, with probability at least $2/3$, the subgraph induced by a set of $\widetilde{O}(1/\epsilon^2)$ vertices of G is $\Omega(\epsilon)$-far from being bipartite.*

Guideline: Following the proof of Lemma 8, note that, for every partition (U_1, U_2) of U, the set S approximates the number of disturbing edges. That is, while the current proof only shows that S hits some disturbing edges, one can actually show that the subgraph induced by S contains $\Omega(\epsilon \cdot |S|^2)$ disturbing edges. Specifically, consider a partition of $\binom{S}{2}$ into $|S|-1$ disjoint perfect matchings, and show that (with high probability) each perfect matching contains $\Omega(\epsilon \cdot |S|)$ disturbing edges.

Exercise 7 (some properties of regular pairs): *Let (A, B) be a γ-regular pair of edge density ρ, and let $\Gamma_B(v) = \{u \in B : \{u, v\}\}$ denote the neighbors of vertex $v \in A$ in the set B. Prove the following claims.*

1. *At least a $1 - 2\gamma$ fraction of the vertices $v \in A$ satisfy $(\rho - \gamma) \cdot |B| \leq |\Gamma_B(v)| \leq (\rho + \gamma) \cdot |B|$.*
2. *If $\rho \geq 2\gamma$, then at least a $(1-2\gamma)^2$ fraction of the vertex pairs $v_1, v_2 \in A$ satisfy $(\rho^2 - 2\gamma) \cdot |B| \leq |\Gamma_B(v_1) \cap \Gamma_B(v_2)| \leq (\rho^2 + 2\gamma) \cdot |B|$.*

Guideline: For Item 1, consider the set of vertices v that violate the degree bound, and focus on the majority that violate the bound in the same direction. For Item 2, fix any vertex v_1 that satisfies Item 1 and let $B' = \Gamma_B(v_1)$.

Exercise 8 (regular pairs in a random graph): *Let A and B be disjoint sets of size N. Prove that a random bipartite graph between A and B is γ-regular with probability at least $1 - \exp(-\gamma^4 \cdot N^2 + 2N)$.*

Guideline: Fixing any $A' \subseteq A$ and $B' \subseteq B$, the probability that $|d(A', B') - d(A, B)| > \gamma$ is exponentially vanishing in $\gamma^2 \cdot |A'| \cdot |B'|$.

Exercise 9 (on the proof of the regularity lemma): *In continuation to Footnote 35, consider the potential function that assigns the partition $(V_1, ..., V_t)$, of $[k]$, the value $k^{-2} \cdot \sum_{i<j} f(V_i, V_j)$, where $f(A, B) = |A| \cdot |B| \cdot d(A, B)^2$.*

1. *Prove that this function does not decrease under a refinement of the partition.*
2. *Prove that if (V_i, V_j) is not γ-regular, then V_i and V_j can be 2-partitioned, into $(V_{i,1}, V_{i,2})$ and $(V_{j,1}, V_{j,2})$, respectively, such that $\sum_{\sigma,\tau \in \{1,2\}} f(V_{i,\sigma}, V_{i,\tau}) \geq f(V_i, V_j) + \gamma^4 \cdot |V_i| \cdot |V_j|$.*

Guideline: For Part 1, consider an arbitrary 2-partition of V_i, denoted (V_i', V_i''), and show that $f(V_i', V_j) + f(V_i'', V_j) \geq f(V_i, V_j)$. Specifically, consider a random variable Z that is assigned $d(V_i', V_j)$ with probability $|V_i'|/|V_i|$ and $d(V_i'', V_j)$ otherwise; note that $\mathbb{E}[Z] = d(V_i, V_j) = \sqrt{f(V_i, V_j)/(|V_i| \cdot |V_j|)}$ whereas $\mathbb{E}[Z^2] = (f(V_i', V_j) + f(V_i'', V_j))/(|V_i| \cdot |V_j|)$; and conclude by using $\mathbb{E}[Z]^2 \leq \mathbb{E}[Z^2]$.

For Part 2, use the subsets $V_i' \subset V_i$ and $V_j' \subset V_j$ that witness the violation of the regularity condition (i.e., satisfy $|d(V_i', V_j') - d(V_i, V_j)| > \gamma$), and consider an analogous random variable Z (which selects one of the four relevant pairs).

Exercise 10 (the number of copies of K_{t_1,t_2} in a dense graph):[72] *Prove that if a k-vertex graph has at least ϵk^2 edges, then it contains at least $\Omega((2\epsilon)^{t_1 t_2}) \cdot k^{t_1+t_2}$ copies of K_{t_1,t_2}* (i.e., the biclique with t_1 vertices on one side and t_2 vertices on the other side).

Guideline: Let $G = ([k], E)$ have degree sequence $d_1, ..., d_k$. Then, $\mathbf{Pr}_{v,u_1,...,u_t \in [k]}[(\forall i \in [t])\ \{v, u_i\} \in E]$ equals $\frac{1}{k} \cdot \sum_{i\in[k]}(d_i/k)^t \geq (\frac{1}{k} \cdot \sum_{i\in[k]} d_i/k)^t = (2|E|/k^2)^t$. Define an auxilary bipartite graph in which the t-subset U is connected to $v \notin U$ if for every $u \in U$ it holds that $\{v, u\} \in E$. Then, the average degree of t-subsets is at least $p \stackrel{\rm def}{=} (2|E|/k^2)^t - \binom{t+1}{2}/k$, where the second term accounts for $\mathbf{Pr}_{v,u_1,...,u_t \in [k]}[|\{v, u_1, ..., u_t\}| < t+1]$. Show that the probability that a random U is connected to t' random v_i's is at least $p^{t'}$.

Exercise 11 (subsets of regular pairs): *Let (A, B) be a γ-regular pair, and $A' \subseteq A$ and $B' \subseteq B$. Prove that (A', B') is a $t \cdot \gamma$-regular pair for $t = \max(2, |A|/|A'|, |B|/|B'|)$.*

Guideline: Note that $C'' \subseteq C' \subseteq C$ satisfies $\frac{|C''|}{|C|} = \frac{|C''|}{|C'|} \cdot \frac{|C'|}{|C|}$.

References

[1] N. Alon. Testing subgraphs of large graphs. *Random Structures and Algorithms*, Vol. 21, pages 359–370, 2002.

[2] N. Alon, E. Fischer, M. Krivelevich and M. Szegedy. Efficient Testing of Large Graphs. *Combinatorica*, Vol. 20, pages 451–476, 2000.

[3] N. Alon, E. Fischer, I. Newman, and A. Shapira. A Combinatorial Characterization of the Testable Graph Properties: It's All About Regularity. In *38th STOC*, pages 251–260, 2006.

[4] N. Alon and M. Krivelevich. Testing k-Colorability. *SIAM Journal on Disc. Math.*, Vol. 15 (2), pages 211-227, 2002.

[5] N. Alon and A. Shapira. Testing subgraphs in directed graphs. *JCSS*, Vol. 69, pages 354–482, 2004.

[6] N. Alon and A. Shapira. Every Monotone Graph Property is Testable. In *37th STOC*, pages 128–137, 2005.

[7] N. Alon and A. Shapira. A Characterization of the (natural) Graph Properties Testable with One-Sided. In *46th FOCS*, pages 429–438, 2005.

[8] N. Alon and A. Shapira. A Characterization of Easily Testable Induced Subgraphs. *Combinatorics Probability and Computing*, Vol. 15, pages 791–805, 2006.

[9] N. Alon and A. Shapira. A Separation Theorem in Property Testing. *Combinatorica*, Vol. 28 (3), pages 261–281, 2008.

[10] N. Alon and J.H. Spencer, *The Probabilistic Method*, John Wiley & Sons, Inc., 1992. Third edition, 2008.

[72] Based on a result in [1].

[11] L. Avigad and O. Goldreich. Testing Graph Blow-Up. In *Studies in Complexity and Cryptography*, pages 156–172, 2011.

[12] M. Bellare, O. Goldreich, and M. Sudan. Free Bits, PCPs and Non-approximability – Towards Tight Results. *SIAM Journal on Computing*, Vol. 27, No. 3, pages 804–915, June 1998.

[13] M. Bender and D. Ron. Testing acyclicity of directed graphs in sublinear time. *Random Structures and Algorithms*, pages 184–205, 2002.

[14] A. Bogdanov and F. Li. A better tester for bipartiteness? arXiv:1011.0531v1 [cs.DS], 2010.

[15] A. Bogdanov and L. Trevisan. Lower Bounds for Testing Bipartiteness in Dense Graphs. In *IEEE Conference on Computational Complexity*, pages 75–81, 2004.

[16] B. Bollobas, P. Erdos, M. Simonovits, and E. Szemeredi. Extremal graphs without large forbidden subgraphs. *Annals of Discrete Mathematics*, Vol. 3, pages 29–41, 1978.

[17] C. Borgs, J. Chayes, L. Lovász, V.T. Sós, B. Szegedy, and K. Vesztergombi. Graph limits and parameter testing. *38th ACM Symposium on the Theory of Computing*, pages 261–270, 2006.

[18] R. Canetti, G. Even and O. Goldreich. Lower Bounds for Sampling Algorithms for Estimating the Average. *IPL*, Vol. 53, pages 17–25, 1995.

[19] E. Fischer and A. Matsliah. Testing graph isomorphism. In *17th SODA*, pages 299–308, 2006.

[20] E. Fischer, A. Matsliah, and A. Shapira. Approximate hypergraph partitioning and applications. In *of 48th FOCS*, pages 579–589, 2007.

[21] E. Fischer and I. Newman. Testing versus estimation of graph properties. In *37th STOC*, pages 138–146, 2005.

[22] O. Goldreich. *Foundation of Cryptography – Basic Tools.* Cambridge University Press, 2001.

[23] O. Goldreich. *Computational Complexity: A Conceptual Perspective.* Cambridge University Press, 2008.

[24] O. Goldreich. Introduction to Testing Graph Properties. In [25].

[25] O. Goldreich (ed.). *Property Testing: Current Research and Surveys.* Springer, LNCS, Vol. 6390, 2010.

[26] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, pages 653–750, July 1998. Extended abstract in *37th FOCS*, 1996.

[27] O. Goldreich, M. Krivelevich, I. Newman, and E. Rozenberg. Hierarchy Theorems for Property Testing. *ECCC*, TR08-097, 2008. Extended abstract in the proceedings of *RANDOM'09*.

[28] O. Goldreich and L.A. Levin. A hard-core predicate for all one-way functions. In the proceedings of *21st ACM Symposium on the Theory of Computing*, pages 25–32, 1989.

[29] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, pages 302–343, 2002. Extended abstract in *29th STOC*, 1997.

[30] O. Goldreich and D. Ron. Algorithmic Aspects of Property Testing in the Dense Graphs Model. *SIAM Journal on Computing*, Vol. 40, No. 2, pages 376–445, 2011.

[31] O. Goldreich and D. Ron. On Proximity Oblivious Testing. *SIAM Journal on Computing*, Vol. 40, No. 2, pages 534–566, 2011. Extended abstract in *41st STOC*, 2009.

[32] O. Goldreich and D. Ron. On Sample-Based Testers. In *6th Innovations in Theoretical Computer Science*, pages 337–345, 2015.

[33] O. Goldreich and I. Shinkar. Two-Sided Error Proximity Oblivious Testing. *ECCC*, TR12-021, 2012. (See Revision 4, 2014.)

[34] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. *Random Structures and Algorithms*, Vol. 23 (1), pages 23–57, August 2003.

[35] M. Gonen and D. Ron. On the Benefit of Adaptivity in Property Testing of Dense Graphs. In *Proc. of RANDOM'07*, LNCS Vol. 4627, pages 525–539, 2007. *Algorithmica* (special issue for RANDOM and APPROX 2007), Vol. 58 (4), pages 811–830, 2010.

[36] T. Gowers. Lower bounds of tower type for Szemeredi's uniformity lemma, *GAFA*, Vol. 7, pages 322–337, 1997.

[37] Håstad, J. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, Vol. 182, pages 105–142, 1999. (Preliminary Version in *28th STOC*, 1996 and *37th FOCS*, 1996.)

[38] D. Hochbaum (ed.). *Approximation Algorithms for NP-Hard Problems.* PWS, 1996.

[39] T. Kaufman, M. Krivelevich, and D. Ron. Tight Bounds for Testing Bipartiteness in General Graphs. *SIAM Journal on Computing*, Vol. 33 (6), pages 1441–1483, 2004.

[40] L.A. Levin. One-way functions and pseudorandom generators. In proc. of the *17th ACM Symposium on the Theory of Computing*, pages 363–365, 1985.

[41] L. Lovász and N. Young. Lecture notes on evasiveness of graph properties. Technical Report TR–317–91, Princeton University, Computer Science Department, 1991.

[42] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, Vol. 20 (2), pages 165–183, 2002.

[43] M. Parnas, D. Ron, and R. Rubinfeld. Tolerant Property Testing and Distance Approximation. *Journal of Computer and System Sciences*, Vol. 72 (6), pages 1012–1042, 2006.

[44] V. Rodl and R. Duke. On graphs with small subgraphs of large chromatic number. *Graphs and Combinatorics*, Vol. 1, pages 91–96, 1985.

[45] D. Ron. Algorithmic and Analysis Techniques in Property Testing. *Foundations and Trends in TCS*, Vol. 5 (2), pages 73–205, 2010.

[46] E. Szemeŕedi. Regular partitions of graphs. In *Proceedings, Colloque Inter. CNRS*, pages 399–401, 1978.