Randomized Algorithms 2019A – Final (Take-Home Exam)

Robert Krauthgamer and Moni Naor

February 25, 2019
Due within 72 hours

General instructions. The exam has 2 parts.

Policy: You may consult textbooks and the class material (lecture notes and homework), but no other sources (like web search). You should work on these problems and write up the solutions by yourself with no help from others.

You may use without proof theorems stated in class, provided you state the appropriate theorem that you are using. As usual, assume n (or $|V|$) is large enough.

Part I (25 points)

Answer 2 of the following 3 questions. Give short answers, that sketch the proof or provide a convincing justification in 2-5 sentences, even for true/false questions.

A. A *triangle* in a graph is just a 3-cycle subgraph. Notice that two triangles can be edge-disjoint even if they share a vertex.

   Is it true that K_n, the complete graph on n vertices, contains $p = \Omega(n^2)$ triangles that are pairwise *edge-disjoint* (i.e., every edge of K_n belongs to at most one triangle)?

   Hint: Can you add a triangle at random?

B. Fix a vertex set $V = \{1, \dots, n\}$ for large enough n. For a parameter $1 < k < n$, let $G^{(k)}$ be the graph on V that is formed by placing a k-clique on vertices $\{1, \dots, k\}$, an $(n-k)$-clique on vertices $\{k+1, \dots, n\}$, and an edge $(k, k+1)$ that connects the two cliques.

   Let $C_{1n}^{(k)}$ be the commute time in $G^{(k)}$ between vertices 1 and n. Is it true that $C_{1n}^{(2)} < C_{1n}^{(n/2)}$?

C. Consider the following randomized algorithm for 2-coloring a bipartite graph: Start with an arbitrary 2-coloring, and while it is not proper, pick any violating edge (its endpoints have the same color) and flip the color of a *random* endpoint of this edge. Provide an upper bound on the expected time (number of flips) it takes to properly color the graph.

Part II (75 points)

Answer 3 of the following 5 questions.

1. Consider a network of n processors arranged in a line (with one node at the end of the line considered 'start' and the other considered 'end'). Each node receives an id between 1 and n, and the goal is to check that the id's constitute a permutation on $1, \dots, n$. The protocol works by the 'start' node sending a message to its neighbor, which in turn sends a message to

its other neighbor and so on, until a message is reached at the 'end' node, which determines the outcome.

Suggest a protocol that works with messages of length $O(\log n)$ bits and reports the correct outcome with probability at least $1-1/n$. For full credit, your protocol should require private (rather than shared) randomness.

2. Consider a clique on n vertices, where initially only one vertex is "informed" (knows some bit b). At each round, every *informed* vertex contacs a random neighbor (among all $n-1$ neighbors) and informs that neighbor; non-informed vertices do nothing. Show that after $O(\log n)$ rounds, with high probability all vertices are informed.

   Hint: Split the analysis into phases. Initially, track the informed vertices. Later, track the non-informed vertices.

3. Recall that in edge-sparsification of hypergraphs, seen in class, $H=(V,E,w)$ is a hypergraph with edge weights $w: E \mapsto \mathbb{R}_+$, and every (nontrivial) $S \subset V$ defines a cut $\delta_H(S) \subset E$. Now define the *product-cost* of this cut to be

$$\operatorname{cost}_H(S) := \sum_{e \in \delta_H(S)} w(e) \cdot |e \cap S| \cdot |e \cap \bar{S}|.$$

   Note the difference from the weight of a cut $w_H(S) = \sum_{e \in \delta_H(S)} w(e)$ that we used in class.

   Show that for every hypergraph $H=(V,E,w)$ and every $\varepsilon \in (0,1/2)$, there is a hypergraph $H'=(V,E',w')$ on the same vertex set, that has at most $(|V|/\varepsilon)^{O(1)}$ edges and is a *product-cost sparsifier* in the sense that

$$\forall S \subset V, \qquad \operatorname{cost}_{H'}(S) \in (1 \pm \varepsilon) \operatorname{cost}_H(S).$$

   Hint: Bound the total sensitivity for product-cost by using the total sensitivity seen in class for weight of a cut.

4. Design a randomized algorithm that, given as input $w_1, \ldots, w_n \geq 0$ and an accuracy parameter $\varepsilon \in (0,1)$, the algorithm stores only $O(1/\varepsilon^2)$ memory words, and then estimates the sum of every subset $S \subset [n]$ by producing for it an estimator Z_S that satisfies

$$\forall S \subset [n], \qquad \Pr\Big[Z_S \in \sum_{i \in S} w_i \pm \varepsilon W\Big] \geq 3/4,$$

   where $W = \sum_{i \in [n]} w_i$ is the sum of all input numbers. Note that the algorithm decides what to store *before* knowing S.

   Hint: First show that storing W and a coordinate $i^* \in [n]$ chosen at random according to probabilities $(\frac{w_1}{W}, \ldots, \frac{w_1}{W})$, can produce estimates $\hat{w}_1, \ldots, \hat{w}_n \in \{0, W\}$ where each $\mathbb{E}[\hat{w}_i] = w_i$.

5. Recall that we used the method of compression to analyze Cuckoo Hashing. Use the method to show that with high probability, a random graph $G_{n,1/2}$ does not contain a clique or independent set of size larger than $O(\log n)$.

Good Luck.