Randomized Algorithms 2024/5
Notes on Homework Set 1

Moni Naor

Kings in Tournaments: Every tournament contains at least one king, the natural way of showing it, given this exercise, is by induction. But an alternative way is to show that the vertex with the maximum outdegree is necessarily a king. Some tournaments contain just one king, e.g. the transitive tournament, whereas other tournaments contain many kings.

Question: show that in a random tournament where every edge is directed independently all the nodes are kings with high probability.

Part (3) of the exercise asked to show that the randomized complexity of finding a king is linear in $|V| = n$. The hinted algorithm was to choose uniformly at random $v \in_R V$ and continue recursively with $C(v)$.

What about a deterministic algorithm? The issue is that we cannot find in a deterministic manner a node with high outdgree with a few probes. The best known algorithm picks $\sqrt{n}$, finds the highest degree in the induced subgraph (taking $O(n)$ queries) and continues with it (its outdegree in the graph is $\Omega(\sqrt{n})$. So $\sqrt{n}$ nodes are cut for the price of $O(n)$. Overall this give an $O(n^{1.5})$ algorithm. The best known lower bound is $\Omega(n^{4/3})$ (it is not so simple). Closing the gap would be very interesting. In any case, this is an example where randomized algorithms have a provable advantage.

What about finding *all* the kings? It turns out you can find three kings (if they exist) for the same cost as one, but finding four kings takes $\Omega(n^2)$ queries even for a randomized algorithm. You can find this and more in a recent paper [1].

About LLMs, I am curious to see what they would especially in question (3), since the linear time algorithm, while very simple and straightforward, was published only very recently in 2024 [1] and presumably the LLMs where not trained on it. So the issue is whether LLMs can deduce new algorithms when given hints.

Number of s-t cuts: An example with an exponential number of s-t cuts can be obtained, for instance, by taking nodes s and t adding a subset U of size $n-2$ and adding edges from s and from t to all nodes in U and no other edge. The minimum s-t cut is of size $n-2$ and by taking any subset $C \subseteq U$ the cut of $\{s\} \uplus C$ vs. the rest is of the this size. There are 2^{n-2} such cuts.

Alternative way of choosing the edges in Karger's algorithm: For an edge $e = (u, v)$ where the degrees of u and v are d_v and d_u respectively we have that the probability of choosing e in the suggested procedure is

$$\frac{1}{n}\left(\frac{1}{d_v}+\frac{1}{d_u}\right).$$

This is not equal to $1/|E|$ in general (an exception is a regular graph). But it is good enough for Karger's algorithm. The key point is the bound on the probability of hitting the min cut is still the same: the min cut is bounded from below by the minimum degree, so the probability of hitting is it still at most $1 - 2/n$.

References

[1] Amir Abboud, Tomer Grossman, Moni Naor and Tomer Solomon, *From Donkeys to Kings in Tournaments*, ESA 2024.
`https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2024.3`