Randomized Algorithms
Homework Set 1

Due Date: Nov. 21st 2024

Moni Naor

A. A tournament is a directed graph (V, E) on a set of nodes V where for every $u, v \in V$ where $u \neq v$ we have that exactly one of the directed edges (u, v) or (v, u) is in E. For a subset $V' \subset V$ we call the tournament (V', E') where for every $u, v \in V'$ we have edge $(u, v) \in E'$ iff $(u, v) \in E$ *the induced tournament on* V'.

A **king** in a tournament is a node $v \in V$ where for every other node $u \in V$ there is a directed path from v to u of length at most two. That is, to be a king it should be possible to get with at most two hops to any other node.

1. For $v \in V$ let $D(v)$ be the set of nodes u where the edge (v, u) exists and $C(v)$ be those where (u, v) exists (i.e. v dominates all those in $D(v)$ and is dominated by those in $C(v)$).

   Show that if $w \in C(v)$ is a king in the tournament induced by $C(v)$, then w is also a king in the full tournament (V, E).

2. Argue that any tournament (V, E) contains at least one king.

3. Suggest a linear in $|V|$ expected time randomized algorithm for finding a king in a tournament. Assume that the tournament is in matrix form for and for $u, v \in V$ checking whether there is an edge (u, v) or (v, u) takes one operation. Can you think of a $o(n^2)$ deterministic one?

4. Pick your favorite LLM. Ask it this question (not including this item) and evaluate its solution. Summarize in one paragraph how well the LLM answered each of the items (1),(2),(3), which LLM and whether it was correct, complete, well explained, concise, misleading,hallucinating, etc.

B. Karger's algorithm also showed a bound on the number of min-cuts, since for every min-cut the probability of outputting this specific cut was at least $2/n^2$. In contrast, show that for s-t cuts, where there are two special nodes s and t that should be separated by the cut, there can be exponentially many min-cuts.

C. In the min-cut algorithm we saw, What happens if the edges are sampled by picking a node $u \in V$ uniformly at random and then picking a random neighbor v of u and outputting the edge (u, v); if there are parallel edges, then the neighbor gets the appropriate weight.

1. Show that this is not identical to picking an edge uniformly at random from E.
2. What can you say about the probability of success of the algorithm when edges are chosen this way?