Exercise 8: Don't get Lost

Task 1: ... everything is (probably) going to be fine $(2 + 3 + 2 + 2)$

An event occurs *with high probability (w.h.p.),* if its probability is, for any choice of $c \in \mathbb{R}_{\geq 1}$, at least $1 - n^{-c}$. Here n is the input size (in our case, $n = |V|$), and c is a (user-provided) parameter, very much like the ϵ in a $(1 + \epsilon)$-approximation algorithm.

This exercise shows nice properties of "w.h.p.", especially why it works so easily under composition.

Algorithm 1 Code for generating a random ID at node v.

1: $\mathrm{id}_v \leftarrow \lceil c \log n \rceil$ random bits from independent, fair sources

a) Suppose that some algorithm $\mathcal{A}$ is called ten times, and each call succeeds w.h.p. Pick c such that for $n \geq 10$, all ten calls of $\mathcal{A}$ all succeed with a probability of at least 0.999.

   Hint: Union bound.

b) Let $\mathcal{E}_1, \ldots, \mathcal{E}_k$ be polynomially many events, i.e., $k \in n^{\mathcal{O}(1)}$, each of them occurring w.h.p. Show that $\mathcal{E} := \mathcal{E}_1 \cap \cdots \cap \mathcal{E}_k$, the event that all $\mathcal{E}_i$ happen, occurs w.h.p.

c) Consider Algorithm 1, which generates random node IDs. Fix two distinct nodes $v, w \in V$ and show that w.h.p., they have different IDs.

d) Show that w.h.p., Algorithm 1 generates pairwise distinct node IDs.

Task 2: ... in the Steiner Forest! $(3 + 3 + 3 + 3 + 2)$

In this exercise, we're going to find a 2-approximation for the Steiner Tree problem on a weighted graph $G = (V, E, W)$, as defined in an earlier exercise; we use the Congest model. Denote by T the set of nodes that need to be connected, and by $G_T = (T, \binom{T}{2}, W_T)$ the terminal graph.

a) For each node v, denote by $t(v)$ the closest node in T. Show that all $v \in V$ can determine $t(v)$ along with the weighted distance $\mathrm{dist}(v, t(v))$ in

$$\max_{v \in V}\{\mathrm{hop}(v, t(v))\} + \mathcal{O}(D)$$

   rounds,[1] where $\mathrm{hop}(v, t(v))$ denotes the hop length of the minimum-weight distance path from v to $t(v)$.

   Hint: This essentially is a single-source Moore-Bellman-Ford with a virtual source connected to all nodes in T.

b) Consider a terminal graph edge $\{t(v), t(w)\}$ "witnessed" by G-neighbors v and w with $t(v) \neq t(w)$, i.e., v and w know that $\mathrm{dist}(t(v), t(w)) \leq \mathrm{dist}(t(v), v) + W(v, w) + \mathrm{dist}(w, t(w))$. Show that if there are no such v and w with $\mathrm{dist}(t(v), t(w)) = \mathrm{dist}(v, t(v)) + W(v, w) + \mathrm{dist}(w, t(w))$, then $\{t(v), t(w)\}$ is not in the MST of G_T!

   Hint: Observe that G is partitioned into Voronoi cells $V_t = \{v \in V \mid t(v) = t\}$, and that in the above case any shortest $t(v)$-$t(w)$ path must contain a node u with $t(u) \notin \{t(v), t(w)\}$, i.e., cross a third Voronoi cell. Conclude that $\{t(v), t(w)\}$ is the heaviest edge in the cycle $(t(v), t(u), t(w), t(v))$.

[1] These are partial shortest-path trees rooted in each $t \in T$.

c) Show that an MST of G_T can be determined and made globally known in $\mathcal{O}(|T|+D)$ additional rounds.

Hint: Use the distributed variant of Kruskal's algorithm from the lecture.

d) Show how to construct a Steiner Tree of G of at most the same weight as the MST of the terminal graph in additional $\max_{v\in V}\{\text{hop}(v,t(v))\}$ rounds.

Hint: Modify the previous step so that the "detecting" pair v, w with $\text{dist}(t(v), t(w)) = \text{dist}(v, t(v)) + W(v, w) + \text{dist}(w, t(w))$ is remembered. Then mark the respective edges $\{v, w\}$ and the leaf-root-paths from v to $t(v)$ and w to $t(w)$ for inclusion in the Steiner Tree.

e) Conclude that the result is a 2-approximate Steiner Tree. What is the running time of the algorithm?

Hint: Recall Task 2 from Exercise 6.

Task 3*: Be more Constructive! $(1 + 1 + 2 + 1 + 2 + 1)$

a) Check up on the prime number theorem!

b) Show that for any $k \in \mathbb{N}$ and any constant $C \in \mathbb{N}$, the number of primes in the range $[2^k, 2^{k+C}]$ is in $2^{\Theta(k+C)}/k$.

c) Prove that for an N-bit number, the number of different $\Theta(\log N)$-bit primes that divides it is bounded by $\Theta(N/\log N)$. Use this to find suitable choices of k and C such that the number of primes in the range $[2^k, 2^{k+C}]$ is polynomial in N and the probability that, for a fixed N-bit number, a uniformly random prime from this range divides it is at most $N^{-\Theta(1)}$.

d) Check up on the AKS primality test!

e) Infer that there is a protocol solving equality with error probability $N^{-\Theta(1)}$ that uses private randomness, communicates $\mathcal{O}(\log N)$ bits, and requires only polynomial computations, both for construction and execution!

f) Check up on your ability to explain this to others in the exercise session!