Exercise 8: Don't get lost...

We always consider connected, simple, weighted graphs $G = (V, E, W)$, where $W : E \to \{1, \dots, n^{\mathcal{O}(1)}\}$, and restrict message size to $\mathcal{O}(\log n)$ bits.

Task 1: ...taking shortcuts!

We sample each node into $S \subseteq V$ with probability $c \log n / n^{1-\alpha}$, for a sufficiently large constant c and some constant $0 < \alpha < 1$. Define the *skeleton graph* (S, E_S, W_S) by having an edge $\{s, t\}$ if s and t are within $n^{1-\alpha}$ hops of each other, and set $W_S(\{s, t\})$ to the minimum weight of an s-t path of at most this number of hops.

It is ok to be brief in this exercise, in particular for arguments that carry over one-to-one from the lecture!

a) Show that $|S| \in \Theta(n^\alpha \log n)$ w.h.p.!

b) Denote by $d_S(s, t)$ the distance of s and t in the skeleton graph. Show that $d_S(s, t) = d(s, t)$ w.h.p.! (Hint: Select for each pair $s, t \in S$ a shortest path from s to t in G. Then prove that a path of the same weight exists in the skeleton graph w.h.p. and use the union bound.)

c) Determine the skeleton graph edges and a $(1 + \varepsilon)$-approximation to their weights, so that each node in S knows its incident edges and their approximate weights. Use $\mathcal{O}((n^{1-\alpha} + |S|) \log n)$ rounds! (Hint: "Find" the edges using the rounding technique from the lecture and the algorithm for the unweighted case. Add a hop counter to each distance that "remembers" the number of traversed edges in the graph, so that paths of more than n^α hops can simply be discarded.)

d) Make the skeleton graph and its approximate weights known to all nodes in $\mathcal{O}(|S|^2 + D)$ rounds.

e) In G, let each node learn its closest $n^{1-\alpha}$ nodes with respect to the distances $\tilde{d}$ (as defined in Theorem 8.20, but for distance parameter $n^{1-\alpha}$), using $\mathcal{O}(n^{1-\alpha} \log n)$ rounds. Show that w.h.p., for each node this list contains a node from S. Define s_v as the closest such node to v with respect to $\tilde{d}$ (breaking ties by identifiers).

f) Argue that in this time bound, routing tables and small labels for routing between v and s_v and vice versa (for all v) can be constructed.

g) Show that if $\tilde{d}(v, s_v) \leq \tilde{d}(v, w)$, then $d(v, s_v) + d(s_v, s_w) + d(s_w, w) \in (7 + \mathcal{O}(\varepsilon)) d(v, w)$. (Hint: Use the technique from the lecture, but make sure to switch to arguing about real distances before applying the triangle inequality!)

h) Conclude that, w.h.p., now all necessary information to route/estimate distances with approximation factor $7 + \mathcal{O}(\varepsilon)$ has been collected. (Hint: Observe that if $\tilde{d}(v, w) < \tilde{d}(v, s_v)$, then you can route on a shortest path. Otherwise use the route from g)!)

i) Sum up the running time bounds and choose an α that minimizes the running time (up to logarithmic factors). What running time do you get?

Task 2: ...in the Steiner Forest!

In this exercise, we're going to solve the Steiner Tree problem, as defined in an earlier exercise. Denote by T the set of nodes that need to be connected.

a) For each node v, denote by t_v the closest node in T. Show that partial shortest-path trees rooted at $t \in T$ spanning the nodes v with $t_v = t$ can be computed in $\max_{v \in V}\{h(v, t_v)\} + \mathcal{O}(D)$ rounds, where denotes the minimum hop length of a shortest path from v to t_v. (Hint: Essentially, this is single-source Bellman-Ford with a virtual source connected to all nodes in T.)

b) Consider the potential terminal graph edges "witnessed" by neighbors v and w with $t_v \neq t_w$, i.e., v and w know that $d(t_v, t_w) \leq d(v, t_v) + W(v, w) + d(w, t_w)$. Show that if there are no such v and w with $d(t_v, t_w) = d(v, t_v) + W(v, w) + d(w, t_w)$, then the terminal graph edge $\{t_v, t_w\}$ is not in the MST of the terminal graph! (Hint: Observe that then any shortest s-t path must contain a node u with $t_u \notin \{s, t\}$. Conclude that $\{s, t\}$ is the heaviest edge in the cycle (s, t_u, t, s).)

c) Show that the MST of the terminal graph 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 $\max_{v \in V}\{h(v, t_v)\}$ rounds. (Hint: Modify the previous step so that the "detecting" pair v, w with $d(t_v, t_w) = d(v, t_v) + W(v, w) + d(w, t_w)$ 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*: ...looking for Thorup and Zwick!

a) Learn about the routing scheme by Thorup and Zwick!

b) Can you see how to use this to speed up the construction from Task 1? Increasing the approximation factor a bit is ok. (Hint: Operate on a skeleton of $\sqrt{n}$ nodes and handle all communication via a BFS tree of G. Close-by nodes are handled as before.)

c) Talk about it in the exercise session!