HW 10

Due on **Wednesday, November 25, 2020 at 10am**

CS/ECE 374: Algorithms & Models of Computation, Fall 2020 Version: **1.1**

Submission instructions as in previous homeworks.

28 (100 PTS.) **OLD Homework problem (not for submission):**
Network deployment.

Consider an undirected connected graph G that represents a map. Every vertex is a location, and every edge represents a road that connects the two locations, with the positive weight of the edge being the distance between the two locations. Let $d(x, y)$ denote the shortest path distance between any two vertices $x, y \in \mathsf{V}(\mathsf{G})$. We have a set $X \subseteq \mathsf{V}(\mathsf{G})$ of locations we would like to connect together via a common *connected* network. Computing the optimal such network is surprisingly difficult, but one can build a reasonably good network.

A natural strategy for deploying/building such a network is as follows – order the vertices of X in some order $v_1, v_2, \ldots, v_t$, where $t = |X|$. In the ith step of the deployment, we install a server at v_i. Let u_i be the closest server among $v_1, \ldots, v_{i-1}$ to v_i. We connect v_i to u_i (paying $d(v_i, u_i)$), and continue to the next location. The ***cost*** of the deployment is $\sum_{i=2}^{n} d(v_i, u_i)$.

28.A. (20 PTS.) Consider the graph $\mathsf{G} = (\{1, 2, \ldots, n\}, \{12, 23, \ldots (n-1)n\})$, where the weight of all edges is 1. Show, that there is an ordering over the vertices of G such that the deployment cost of $X = \mathsf{V}(\mathsf{G})$ is $\Omega(n \log n)$.

28.B. (20 PTS.) Let T be the cheapest tree, such that $X \subseteq \mathsf{V}(\mathsf{T})$. Verify that the cost of the tree $w(\mathsf{T}) = \sum_{e \in \mathsf{T}} w(e)$ is a lower bound on the deployment cost of X for any ordering.

Prove that there exists a closed walk that visits all the vertices of X, and the total weight of the edges of the walk is at most $2w(\mathsf{T})$.

28.C. (20 PTS.) Let x, y be the closest pair of vertices in X. Formally, it is one of the pairs realizing $\min_{u \in X} \min_{v \in X \setminus \{u\}} d(u, v)$. Prove, using the above, that $d(x, y) \leq 2w(\mathsf{T}) / |X|$,

28.D. (40 PTS.) Consider the greedy algorithm that computes the closest pair of vertices $x, y \in X$, sets $v_t = x$, and then recursively computes the ordering for $X \setminus \{x\}$. Let Π be the resulting ordering of X. Prove that the deployment cost of X is bounded by $O(w(\mathsf{T}) \log n)$.

(Clearly, this algorithm can be implemented in polynomial time – but the exact running time here is unimportant.)

29 (100 PTS.) **OLD Homework problem (not for submission):**
Few spies.

Let $\mathsf{G} = (C \cup S, \mathsf{E})$ be a graph with n vertices and m edges. The vertices of $C \cup S$ represents citizens in the glorious democratic republic of north Narnia (GDRNN). An edge between two people indicates that they are friends. The vertices of S represents people that are willing to spy on their friends to see if they do any illegal activities (like enjoying themselves, etc). The government of GDRNN would like to choose a set of spies $S' \subseteq S$, of minimum size, such that every citizen in C is connected to some vertex of S' (the members of S are trusted by the government – so no need to spy on them). Computing the smallest such set is surprisingly difficult. Again, we are going to be happy with a simple greedy strategy[1].

[1] Seeing the movie "The lives of others" might not help you solve this problem, but you might enjoy it anyway.

Let $C_0 = C$ and $S_0 = \emptyset$. In the ith iteration, for $i > 0$, we pick the vertex s_i in S that is connected to the largest number of citizens in C_{i-1} (resolving ties arbitrarily). We then set $S_i = S_{i-1} \cup \{s_i\}$, and $C_i = C_{i-1} \setminus \Gamma(s_i)$, where $\Gamma(s_i)$ is the set of citizens connected to s_i. We stop as soon as C_i is empty, and output S_i as the desired set of spies.

29.A. (20 PTS.) Assume that the optimal solution is of size k. Prove, that for any i, s_i is connected to at least $|C_{i-1}|/k$ citizens in C_{i-1} (hint: think about the k optimal spies $o_1, \ldots, o_k$).

29.B. (20 PTS.) Prove that $|C_i| \leq (1 - 1/k)\,|C_{i-1}|$.

29.C. (20 PTS.) Using that $(1 - 1/k)^k \leq 1/e$, prove that for all i, we have that $|C_{i+k}| \leq |C_i|/e$.

29.D. (40 PTS.) Prove, that the above algorithm outputs a set of at most $k(\lceil \ln n \rceil + 1)$ spies, such that all the citizens of C are spied on by these spies.

30 (100 PTS.) **OLD Homework problem (not for submission):**

Undecidable, that's what you are.

For each of the following languages, either prove that it is undecidable (by providing a detailed reduction from a known undecidable language), or describe an algorithm that decides this language – your description of the algorithm should be detailed and self contained. (Note, that you cannot use Rice Theorem in solving this problem.)

30.A. (25 PTS.) $L_1 = \{\langle M, N \rangle \mid L(M) = \overline{L(N)}$, where M is a Turing machine, and N is a $\mathsf{NFA}\}$.

30.B. (25 PTS.) $L_2 = \{\langle N \rangle \mid L(N)$ is infinite, where N is an $\mathsf{NFA}\}$.

30.C. (25 PTS.) $L_3 = \{\langle R, N \rangle \mid L(R) = L(N)$, where R is a regular expression, and N is an $\mathsf{NFA}\}$.

30.D. (25 PTS.)

$$L_4 = \{\langle M \rangle \mid L(M) \text{ contains some word of even length, where } M \text{ is a Turing machine}\}.$$