

CS3100: Algorithms

Homework 5

1. Consider a set of mobile computing clients in a town, where each client needs to be connected to one of several possible base stations. We'll suppose there are n clients, with the position of the client specified by its (x, y) coordinates in the plane. There are also k base stations, where the position of each of these is also specified by its (x, y) coordinates.

For each client, we wish to designate exactly one base station to serve its requests. Connections are constrained, however. There is a range parameter, r , and each client can only connect to base stations within distance r . There is also a load parameter L , and no more than L clients can be connected to a single base station.

Your goal is to design a polynomial time algorithm to decide, given the positions of a set of n clients and k base stations, as well as range and load parameters, if every client can be connected simultaneously to a base station without violating the range and load conditions described above.

2. We define the escape problem as follows. We are given a directed graph $G = (V, E)$ (picture a network of roads). A certain collection of these nodes $X \subset V$ are designated as populated nodes, and a certain other collection $S \subset V$ are designated as safe nodes. (You may assume that S and X are disjoint.) In case of an emergency, we want evacuation routes from the populated nodes to the safe nodes. A set of evacuation routes is defined as a set of paths in G so that (i) each node in X is the tail of one path, (ii) the last node in each path lies in S , and (iii) the paths do not share any edges. Such a set of paths gives a way for occupants of the populated nodes to “escape” to S , without overly congesting any edge in G .
 - (a) Given G, X , and S , show how to decide in polynomial time if such a set of evacuation routes exists.
 - (b) No suppose we have the same problem as above, but want to enforce a stronger “no congestion” rule: here, (iii) is changed to say “the paths do not share any nodes”. With this new condition, show how to decide in polynomial time if such a set of routes exist.
 - (c) Finally, provide an example with the same G, S , and X , in which the answer to question (a) is yes but the answer to question (b) is no.

3. You've been called in to help some network administrators diagnose the extent of failure in their network. The network is designed to carry traffic from a source s to a target t , so we'll model it as a directed graph where every edge has capacity 1 and where each node lies in at least one path from s to t .

Now, when everything is running smoothly, the maximum flow has value k . However, the current situation (and the reason you are here) is that an attacker has destroyed some of the edges in the network, so there is no s to t path. The administrators know that the attacker only destroyed k edges, the minimum number needed to separate s and t .

Now the administrators are running a tool on s with the following behavior. If you issue the command $ping(v)$, for a given node v , it will tell you if there is currently a path from s to v . They'd like to determine the extent of failure using this tool, but it's not practical to ping every node in the graph. Therefore, they are coming to you to design an algorithm that reports the full set of nodes not currently reachable from s . You could do this by pinging everyone, but you'd like to do it using many fewer ping commands.

Give an algorithm that accomplishes this task using only $O(k \log n)$ ping commands.