

Homework Assignment #2
Due: Oct 27, 2016, by 7pm

For all algorithms design problems, state how you model the problem (if needed), describe your algorithm in words, then give a pseudocode.

[30] 1. **Maintenance costs**

A telecom company is running its fiber optic network through rural Newfoundland around Gander. With the harsh weather here, they are mainly concerned with the maintenance costs of the network than with the costs of laying the fiber. A local consultant gives them estimates of maintenance cost per year of various possible links between different villages (including Gander).

- (a) State which computational problem they need to solve to find the least costly way of providing a fiber network to these villages.
- (b) Suppose that after they announce their connection scheme, some village offers them to cover some of the costs if it gets a direct link to Gander. Give a linear-time algorithm to decide whether the offer is good enough. The inputs to your algorithm are the previous connection scheme, all the original estimates and the new cost of the link between this village and Gander.
- (c) Now, suppose that after they built the network, there was a flood, and one of the links they built got washed away. To prevent that happening again, they would need to put in extra infrastructure if they rebuild that link, increasing its maintenance cost. Give a linear-time algorithm that determines whether to rebuild the link or to build a new link elsewhere.

[30] 2. **Greedy algorithms**

Consider a long, quiet country road with houses scattered very sparsely along it. (We can picture the road as a long line segment, with a south endpoint and a north endpoint, with locations of all houses mapped on it.) You want to place cell phone base stations at certain points along the road, so that every house is within four kilometers of one of the base stations.

- (a) Give a linear-time algorithm that achieves this goal, using as few base stations as possible. Your input is the list of houses along the road from south to north, where coordinates of every house is just its distance from the south end of the road.
- (b) Now, prove the correctness of your algorithm using induction. That is, define a "promising" partial solution and show that at every step of the algorithm your solution is indeed promising.

[20] 3. **Number of solutions**

Consider the single-source shortest path problem with positive edge weights.

- (a) Give an example of a graph on 3 vertices s, v, t and with three directed edges, where there is more than one shortest path from s to t .
- (b) Give an $O(m \log n)$ algorithm that computes, for each vertex v in the graph, the number of shortest paths from s to v . Hint: modify one of the algorithms we covered in class.

[20] 4. **Shortest path with negative edges**

I have mentioned in class that on a directed acyclic graphs many problems become easier. Let's take advantage of that, in this case for solving single-source shortest path problem with negative weights. Give an algorithm that checks if the input graph is a DAG, and if so, solves single-source shortest-path problem, otherwise outputs "Use Bellman-Ford". The input to your algorithm is G, s, c , where $c: E \rightarrow \mathbb{R}$. The total running time of your algorithm should be $O(m + n)$.