CS 312: Algorithms **Fall 2018**

Discussion 5

Your Name: ____________________ Collaborators: ____

You will be randomly assigned groups to work on these problems in discussion section.

Problem 1. Greedy stays ahead. Suppose that three of your friends have decided to hike the Appalachian Trail this summer. They want to hike as much as possible per day but not after dark. They've identified a large set of good stopping points for camping, and they're considering the following system for deciding when to stop for the day. Each time they come to a potential stopping point, they determine whether they can make it to the next one before nightfall. If they can make it, then they keep hiking; otherwise, they stop.

Despite many significant drawbacks, they claim this system does have one good feature. "Given that were only hiking in the daylight," they claim, "it minimizes the number of camping stops we have to make."

Is this true? The proposed system is a greedy algorithm, and we wish to determine whether it minimizes the number of stops needed.

To make this question precise, let's make the following set of simplifying assumptions.

- The Appalachian Trail is a line segment of length L.
- Your friends can hike d miles per day (independent of terrain, weather conditions, and so forth).
- The potential stopping points are located at distances $x_1, x_2, \ldots, x_n$ from the start of the trail.
- Your friends are always correct when they estimate whether they can make it to the next stopping point before nightfall.

Your friends' greedy algorithm selects a particular set of stopping points. We'll say that a set of stopping points is *valid* if the distance between each adjacent pair is at most d, the first is at distance at most d from the start of the trail, and the last is at distance at most d from the end of the trail. Thus a set of stopping points is valid if one could camp only at these places and still make it across the whole trail. Well assume, naturally, that the full set of n stopping points is valid; otherwise, there would be no way to make it the whole way.

We can now state the question as follows. Is your friends' greedy algorithm—hiking as long as possible each day—optimal, in the sense that it finds a valid set whose size is as small as possible?

Problem 2. Topological Sorting.

Below is a list of edges in a directed graph with nodes A, B, C, D, E, and F.

$$B \to E,\ B \to F,\ C \to D,\ D \to A,\ E \to F,$$

1. Find five topological sortings of the graph above.
2. Say that you want the ordering A, B, C, D, E, F to be a topological ordering. Which edge must be removed to do this?
3. Say that you didn't want any topological orderings to be possible. What is an edge you could add that would achieve this effect.
4. Say that the only nodes in the graph were B, E, and F. How many topological orderings are possible here?