Homework 6

Due Wednesday, March 2nd at 11:59 pm ET on Gradescope

Solution guidelines For problems that require you to provide an algorithm, you must give the following:

1. a precise description of the algorithm in English and pseudocode (*),
2. a proof of correctness,
3. an analysis of the asymptotic running time and space.

You may use algorithms from class as subroutines. You may also use any facts that we proved in class, e.g. correctness of subroutines, running time of subroutines.

You should be as clear and concise as possible in your write-up of solutions.

A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand.

(*) It is fine if the English description concentrates on the high level ideas and doesn't include all the details. But the reader should not have to figure out your solution solely based on the pseudocode. You can also add comments to your pseudocode, in fact that is best practice. FYI we will share a document on good pseudocode style with you and it will also be discussed in labs and lecture.

Problem 1. *Pizza (10 points.)*

You are giving out slices of pizza to friends. However, your friends are very picky, each of them has an exact wedge of the pizza in mind they want. (The pizza arrives at your house without being cut into slices.) Each friend gives you the two angles of their preferred slice. For example one friend might request the wedge 17.23° to 42°, another wants 34° to 51.5° and a third 350° to 2°. (Note that the angles might not be integers, and 0 may be inside a wedge.) Each friend only accepts the entire wedge they asked for or doesn't eat at all. Unfortunately, the requested wedges overlap, so you cannot satisfy all of your friends.

Design an algorithm that given as input the n pairs of angles, selects a maximum non-overlapping subset of slices. Hint: This problem should be solved in $O(n^2)$.

Problem 2. *Ruggles station (10 points.)*

Ruggles station is a train stop in Boston close to many schools and hospitals in the city, yet, very few trains actually stop at this location. Boston is planning to revise train schedules to allow more trains to stop at the Ruggles station. Here is the list of proposed train arrival and departure times at Ruggles during the morning hours. Note, that a train spends the time between its arrival and departure standing at a platform to allow passengers to get on and off.

provider	arrives	departs
MBTA	6:55	7:09
Acela	7:00	7:05
Amtrak	7:10	7:30
Amtrak	7:55	8:10
MBTA	8:00	8:15
Amtrak	8:22	8:46
Amtrak	8:45	9:05
MBTA	8:55	9:10
Acela	9:01	9:10

(a.) Is Ruggles station - which only has *two* platforms - able to accommodate the proposed time table?

Answer yes or no and give a one sentence explanation.

(b.) Boston is also considering other timetables and expanding the station by building new platforms. But building platforms is expensive.

Describe how to use or modify an algorithm we have already seen in class to do the following task. Your algorithm should take as input a timetable with arrival and departure times for trains at some station and outputs the optimal (fewest) number of platforms required for the station's schedule to be satisfied. (The arrival and departure times of trains is given to you in a random order.)

(c.) The Acela Express is the fastest train in the Northeast. Assume it runs *once* a day from Boston to Washington D.C. Because the Acela Express is longer than the other trains only platform 1 in Ruggles station (that is built longer than the others) can accommodate it.

Design an algorithm that takes as input the same (random order) train schedule as in part **(b.)** and assigns trains to the minimum number of platforms. Of course, your algorithm should take care of the Acela train needing platform 1 as well. *Hint: a slight modification of the algorithm for part* **(b.)** *works.*