Spring 2021 CSCI 4020 – Design & Analysis of Algorithms

MIDTERM EXAM

- There are 3 problems not including the Academic Pledge. Prove everything unless otherwise directed. You are allowed to use (without a proof) things we established in class.
- You must submit this exam on LMS by 2:20pm Troy time, late exams will not be accepted. You can use LaTeX to write your solutions, or write them by hand and scan them in, or take pictures with your phone and upload to LMS, etc. As long as we can open the file on LMS and your solution is legible (we can read it), any method you choose is fine.
- Your official time limit for taking the exam is only 100 minutes: at that point you must submit whatever you have written up. After submitting, you can continue working on the exam until the deadline at 2:20pm and submit new versions, but officially the last 20 minutes is reserved for making sure that you have enough time to submit your solutions. It is your responsibility to check that you submitted the correct version of your exam, and to submit early enough that you are able to deal with any technical issues and are able to verify that your submission uploaded successfully. Keep a local copy of the files you submit just in case, as well as the email submission receipt from LMS.
- You may not collaborate, discuss, or provide your exam solutions to anyone, except the course instructor and TAs. While you are allowed to use the class textbook and your notes and handouts (anything posted on LMS) during exams, using any other resources to formulate solutions during the exam, including any material found on the Internet or consulting with anyone else, is a clear breach of academic integrity and will be punished severely. Absolutely no communication with anyone else is allowed during the exam.
- We will not be answering questions during the exam, so do the best you can. If you believe there is a mistake in the problem statement or something is unclear, write a note explaining this as part of your solutions.
- Good luck!

1 Academic Integrity Pledge

Please copy the following statement exactly as it appears, and sign below it. You can just type your name if you like if you are submitting a type-written document. You must sign this statement to get credit for this exam.

"On my honor, I have neither given nor received any unauthorized aid on this exam."

2 Minimum Spanning Trees (20 points)

Decide whether you think each of the following statements is true or false. If it is true, give a short explanation (you do not need to give a full proof). If it is false, give a counterexample.

Part A

Suppose we are given an undirected graph G, with integer edge costs that are all positive and distinct. Suppose P is the unique cheapest path connecting two nodes u and v (all the other paths between u and v are strictly more expensive). Here cost of a path is the sum of costs of all edges in the path.

True or false? Path P must be part of some minimum spanning tree.

Part B

Suppose we are given an undirected graph G, with integer edge costs that are all positive and distinct. Let e and f be edges of G, with f more expensive than e.

True or false? Edge f will never belong to a minimum spanning tree unless e is also in it.

3 Concert Series (20 points)

Imagine that you are a student at the start of a summer vacation. There is a summer concert series near your town, so you decide to spend a lot of your time attending concerts. During each of the n days of your vacation, there is a different concert by a different band. You like some of these bands more than others: let's say that attending the concert on day i gives you f_i amount of fun. Unfortunately, these concerts can be very tiring: after attending a concert on day i you need at least x_i days to rest, and cannot attend any other concerts until this many days have passed. (You cannot attend a concert if there is not enough time left in the summer to finish your rest.)

You must decide, for each day, whether you are going to a concert or not, with the goal of maximizing the total amount of fun during the entire n days.

To summarize: Given nonnegative numbers $f_1, \ldots, f_n$ and $x_1, \ldots, x_n$, give an algorithm to form a schedule saying on which days you will go to a concert. After attending a concert on day i, you cannot attend any more until x_i days have passed, and you cannot attend a concert if not enough days are left in the summer to finish your rest. Your schedule should maximize the total amount of fun that you gain from attending the concerts. The algorithm should run in time at most $O(n^2)$.

Example: Suppose f_i are $4, 4, 4, 9, 4, 25$ and x_i are $1, 1, 1, 2, 1, 5$. Then after going to a concert on day 1, you cannot also go on day 2, since you need a day of rest. A valid solution would be to go to a concert on days 1, 3, and 5, for a total of 12 fun. But this is not the optimum solution: that would be going to a concert on days 1 and 4, giving a total of 13 fun.

Hint: There are several correct approaches to solving this problem. You can set $OPT[i]$ to be the optimum solution given only days $1 \ldots i$, and form a recurrence based on these subproblems. But some people may find it easier to set $OPT[i]$ to be the optimum solution given only the days $i \ldots n$ instead, and form an appropriate recurrence for these subproblems instead.

4 Fishing (20 points)

You are playing a fishing video game. In this simple game there are n days, with x_i fish coming to be caught by you on day i. Unfortunately, how many fish you catch is a function of your *fishing skill*: if your fishing skill is j then you can only catch j. To make things simple, let's just say that if your fishing skill is j on day i, then you will catch exactly $\min(j, x_i)$ fish (since you cannot catch more fish than there are on that day).

Unfortunately, your fishing skill at the start of the game is 0, so no matter how many fish are available, you will not be able to catch any. Fortunately, on any day you can decide that, instead of fishing, you will train in order to improve your skill. After training for one day, your fishing skill increases by 2 permanently, but of course this means that you do not catch any fish on that day since you spent it training instead.

In summary, you are given the numbers $x_1, \ldots, x_n$. Design an algorithm which determines on which days you should go fishing and on which days you should train, so that the number of fish you catch in total is maximized. Your algorithm should run in time $O(n^2)$.

Example: Given the vector of x_i's $[0, 1, 10, 1, 10, 10]$, you can train on the first, second, and fourth day, and fish on the rest. This results in 4+6+6=16 fish in total.

Hint: You will need to use two-dimensional subproblems to solve this.