CS 577-2: Introduction to Algorithms 11/24/2020

Midterm Exam 2

Instructor: Dieter van Melkebeek

Guidelines:

- You can use all the results we showed in class. Clearly state the results you use.
- You are not expected to argue correctness and analyze the complexity of *every* algorithm you design. You only need to provide the elements that are explicitly asked for.
- Pay attention to the Piazza thread "Clarifications on Midterm Exam 2" for potential clarifications.
- If you have a question, ask it on Piazza. During the exam your Piazza posts will only be visible to the instructors. If the instructors feel the response should be made visible for all students, they will add a clarification to the above thread.
- The exam is open-book and open-notes. You are free to use scratch paper.
- You are not allowed to communicate with anyone but the instructors throughout the duration of the exam.
- The exam is proctored by Honorlock. During the exam you are not allowed to use electronic devices other than the computer on which you are taking the exam.
- The only URLs you are allowed to access are `http://www.piazza.com` for Piazza, `https://wisconsin-madison.instructure.com/courses/205209` for Canvas, and `http://jeffe.cs.illinois.edu/teaching/algorithms` for the recommended online textbook. Note that you will need to login to Canvas the first time you access through the links above. If (and only if) you have trouble accessing the links, you may use a browser other than Google Chrome (leave the quiz open in Google Chrome). This will be flagged and reviewed, so do not access links other than those above.
- You may take one 5-minute break maximum. Otherwise, you need to have your face within view of the webcam for the duration of the exam. You cannot wear headphones, earbuds or hats but are welcome to wear earplugs.
- The exam ends at 3:45pm. You need to stop working on the exam no later than that time.
- Your solutions can be typed or handwritten. You have till 3:50pm to upload them in PDF format to Canvas. You are welcome to use your phone or other electronic devices for PDF conversion once you stopped working on the exam.
- Canvas will continue accepting uploads until 4:00pm Anyone who submits between 3:50pm and 4:00pm may only do so if they stopped working by 3:45pm and had issues uploading. We will review the Honorlock videos from 3:45pm onward for any student who submits after 3:50pm.
- Good luck & Happy Thanksgiving!

CS 577-2 Fall 2020 Midterm Exam 2

You are managing a snowboarding resort and run into the following problems.

(a) [7 points] You are fully booked today: there are n participants needing a snowboard, and you have exactly n snowboards. The snowboards have various lengths; $\ell[j]$ denotes the length of the j-th snowboard for $i \in [n]$. Ideally, every participant $i \in [n]$ gets a snowboard of length equal to their height $h[i]$, but that ideal may not be feasible. Instead, your goal is to minimize the total discrepancy, i.e., the sum of the discrepancies over all participants, where the discrepancy of a participant of height h matched with a snowboard of length ℓ equals $(h - \ell)^2$.

For example, suppose $n = 2$, $\ell[1, 2] = [170, 195]$. and $h[1, 2] = [175, 185]$. There are two ways to match the participants to the snowboards:

participant	snowboard	discrepancy
1	1	$(h[1] - \ell[1])^2 = (175 - 170)^2 = 25$
2	2	$(h[2] - \ell[2])^2 = (185 - 195)^2 = 100$

participant	snowboard	discrepancy
1	2	$(h[1] - \ell[2])^2 = (175 - 195)^2 = 400$
2	1	$(h[2] - \ell[1])^2 = (185 - 170)^2 = 225$

The first matching yields a total discrepancy of $25 + 100 = 125$, and the second one of $400 + 225 = 625$. Thus, the first matching is optimal,

Design an algorithm that computes the minimum total discrepancy achievable. Your algorithm should run in time $O(n \log n)$.

Argue correctness and analyze the running time.

(b) [4 points] You employ m instructors and are planning their work for the next period of k days. You are given an $m \times k$ table A where $A[i, j]$ indicates whether instructor i is available to work on day j. For each day $j \in [k]$ you also know the number $d[j]$ of instructors needed that day. Your goal is to come up with a work schedule such that no instructor works on a day they are not available, and each day $j \in [k]$ exactly $d[j]$ instructors work. More precisely, you need to output an $m \times k$ table S where $S[i, j]$ indicates whether instructor i needs to work on day j.

For example, suppose the period has $k = 4$ days, $d[j] = 3$ for each $j \in [k]$, and there are $m = 5$ instructors with the availability given in the table A on the left. Then the table S on the right describes a satisfactory schedule.

A	day 1	day 2	day 3	day 4
instructor 1	✓	✓		
instructor 2	✓		✓	
instructor 3	✓	✓	✓	✓
instructor 4		✓	✓	✓
instructor 5		✓	✓	✓

S	day 1	day 2	day 3	day 4
instructor 1	✓	✓		
instructor 2	✓		✓	
instructor 3	✓		✓	✓
instructor 4		✓		✓
instructor 5		✓	✓	✓

[Problem statement continues on the next page.]

Design an algorithm that finds a schedule meeting all requirements, or reports that none exists. Your algorithm should run in time $O((m+k)mk)$.

Analyze the running time.

(c) [4 points] When you announce your plan, one of the instructors, $i^* \in [m]$, asks to work one day less. You want to update your plan so as to accommodate that request but do not want to increase the maximum number of days that instructors need to work.

In the above schedule S, instructors 1, 2, and 4 each work two days, and instructors 3 and 5 each work 3 days, so the maximum number of days that instructors work is 3.

- Suppose instructor $i^* = 4$ asks to work one day less. There is no way to accommodate that request without increasing the maximum number of days that instructors work to 4.
- Suppose instructor $i^* = 1$ asks to work one day less. You can accommodate that request without having anyone work more than 3 days, namely by letting instructor 3 take over day 2 from instructor 1, and letting instructor 4 take over day 3 from instructor 3.

Develop an algorithm that takes as input an availability table A, a valid schedule S, and an instructor $i^* \in [m]$, and checks whether it is possible to reduce the work load of instructor i^* by one day while maintaining all requirements and without increasing the maximum number of days that instructors work. Your algorithm should run in time $O(mk)$.

Argue correctness.