CS 577-2: Introduction to Algorithms 10/13/2020

Midterm Exam 1

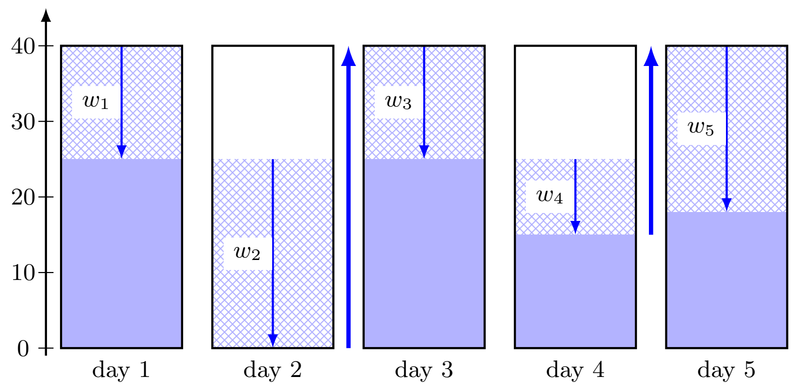
Instructor: Dieter van Melkebeek

Guidelines:

- The exam consists of a single problem setting with three parts. When solving a later part, you can assume a solution for the earlier parts if that is helpful.
- You can use all the results we showed in class. Clearly state the results you use.
- Unlike for the homework, 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 1" for potential clarifications throughout the exam.
- 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 any 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. Only use Google Chrome, and follow the above links directly (right click to open in a new tab) so you don't access other websites (such as Google) in the process.
- 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 3:45. You need to stop working on the exam no later than that time.
- Your solutions can be typed or handwritten. You have till 3:50 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. Anyone who submits between 3:50 and 4 may only do so if they stopped working by 3:45 and had submission issues uploading. We will review the Honorlock videos from 3:45 onward for any student who submits after 3:50.
- Good luck!

You run a chemical lab, and are planning the supply of purified water for a period of n days. You know that you will need $w_i \leq 100$ gallons during day i. To accommodate this, you will acquire a storage tank with a capacity of some c gallons. The tank will be filled to capacity some f days throughout the period, each time at the start of the day. It is always filled to capacity at the start of the first day. It should never happen that the demand for a day cannot be met.

As an example, consider a period of $n = 5$ days with $w_1 = 15$, $w_2 = 25$, $w_3 = 15$, $w_4 = 10$, and $w_5 = 22$. As illustrated below, it works to have $c = 40$ and fills at the start of days 1, 3, and 5.



(a) [3 points] One problem is deciding how large the capacity c of the tank should be for a given bound on the number of fills. You know that an optimal schedule in this setting is to postpone each fill as much as possible without letting the supply run out. Given n, f, and the values w_i for $i \in [n]$, what is the minimum capacity $c \in \mathbb{N}$ so that this schedule leads to no water shortages and at most f fills?

Continuing the example above, for $f = 3$ the figure indicates that $c = 40$ works. If we try $c = 39$, then we need to fill at the start to days 1, 2, 3, and 5. So, the answer is $c = 40$.

Design an algorithm that runs in time $O(n \log n)$. You do *not* need to argue correctness *nor* to analyze the running time.

(b) [7 points] Some of the days a fill would be more disruptive for the lab than others. You have quantified these as a daily loss of revenue ℓ_i that the lab incurs when a fill happens at the start of day i for $i \in [n]$. Given n, c, and the values w_i and ℓ_i for $i \in [n]$, you would like to know how small you can make your total loss of revenue due to the fills without having any supply shortage. There are no constraints on the number f of fills in this problem, and fills do not need to follow the schedule from part (a) but do need to include the first day.

Consider the example above with $c = 40$, $\ell_1 = 0$, $\ell_2 = 1$, $\ell_3 = 100$, $\ell_4 = 20$, and $\ell_5 = 30$. The schedule in the figure costs $\ell_1 + \ell_3 + \ell_5 = 130$ loss in revenue. By instead refilling at the start of days 1, 2, and 4, the loss in revenue is reduced to $\ell_1 + \ell_2 + \ell_4 = 21$. This is optimal, so the answer is 21.

Design an algorithm that runs in time $O(n^2)$ and space $O(n)$.

Argue correctness and analyze the running time.

[Problem statement continues on the next page.]

(c) [4 points] Assume you have an algorithm for part (b) that you can call as a blackbox.

Design an algorithm that retrieves a solution and outputs a fill schedule that achieves the minimum total loss in revenue in (b). Your algorithm can make $O(n)$ calls to the blackbox and spend $O(n)$ time outside of the calls.

For the example from part (b), the output for would be: days 1, 2, and 4.

Analyze the number of calls to the blackbox.

Extra credit: In order to achieve the minimum total loss of revenue, the value of f may need to be larger than the supplier is willing to accommodate. For their choice of f, the supplier would like to know how small they can make your total loss in revenue. Design an algorithm that runs in time $O(fn^2)$ and space $O(n)$.