CS4800: Algorithms — S'18 — Jonathan Ullman

Homework 10
Due Friday Apr 20 at 11:59pm via Gradescope

Name:
Collaborators:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Friday Apr 20 at 11:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typeset in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly prohibited.

Problem 1. *Dunkin' Donuts*

There are n Dunkin' Donuts $D_1, \ldots, D_n$ on I-90 from Boston to Albany. With a cup of coffee, you are able to drive M miles before you get tired and need another cup. Dunkin' Donuts D_1 is in Boston. Each Dunkin' Donuts D_i, for $2 \le i \le n$ is $m_i \le M$ miles past the previous Dunkin' Donuts D_{i-1}. Dunkin' Donuts D_n is in Albany.

Design a greedy algorithm that finds a minimum-size set of stops to make such that you will always be well caffeinated on your drive.

(a) Give pseudocode for your greedy algorithm.

Solution:

(b) Prove that your algorithm finds a set of stops of minimum size.

Solution:

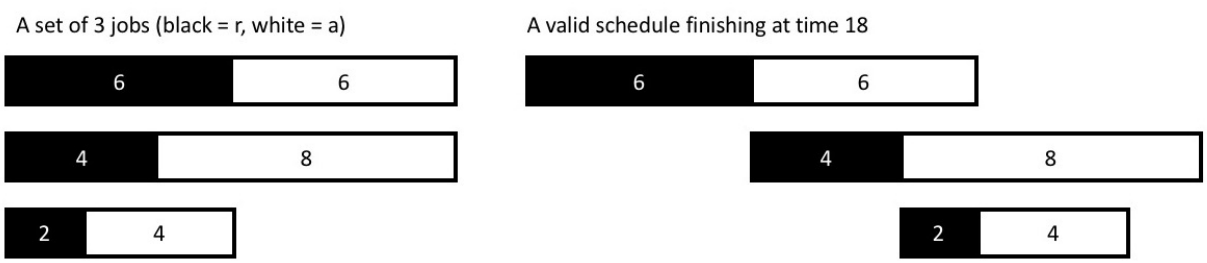
(c) Analyze the running time of your algorithm?

Solution:

Problem 2. *Seating Arrangements*

You are the great Dutch painter Rembrandt Harmenszoon van Rijn, renowned the world over for your prolific portrait work. However, you wouldn't be so prolific without your army of apprentices, and your skill with scheduling algorithms. You've been contracted by a wealthy aristocrat to produce portraits of each of his family members, and to preserve your reputation you must complete all the portraits as quickly as possible. Each portrait requires you to spend some of your time starting it, and then will require some amount of time for one of your interchangeable apprentices to finish. There is only one of you, but there is an infinite number of apprentices who can work in parallel. To satisfy your patron, you need to schedule the work so that the entire batch of portraits are finished as quickly as possible.

Design an algorithm that takes a set of n portraits, each with a pair $p_i = (r_i, a_i)$ where r_i is the amount of time you (Rembrandt) must spend on it and a_i is the amount of time it will take an apprentice to finish with it, and outputs a schedule for the work so that the last apprentice finishes the last portrait at the earliest possible time. An example of a valid input and valid (though not necessarily optimal) solution is given below.



(a) In what order does your greedy algorithm sort the inputs?

Solution:

(b) Give pseudocode for your greedy algorithm.

Solution:

(c) Prove that your algorithm finds an optimal schedule.

Solution:

(d) Analyze the running time of your algorithm?

Solution: