CSCI 332, Fall 2024

Homework 10

Due before class on Tuesday, November 12, 2024–that is, due at 9:30am Mountain Time

---

Submission Requirements

- Type or clearly hand-write your solutions into a PDF format so that they are legible and professional. Submit your PDF on Gradescope.
- Do not submit your first draft. Type or clearly re-write your solutions for your final submission.
- Use Gradescope to assign problems to the correct page(s) in your solution. If you do not do this correctly, we will ask you to resubmit.
- You may work with a group of up to three students and submit ***one single document*** for the group. Just be sure to list all group members at the top of the document. When submitting a group assignment to Gradescope, only one student needs to upload the document; just be sure to select your groupmates when you do so.

Academic Integrity

Remember, you may access ***any*** resource in preparing your solution to the homework. However, you ***must***

- write your solutions in your own words, and
- credit every resource you use (for example: "Bob Smith helped me on this problem. He took this course at UM in Fall 2020"; "I found a solution to a problem similar to this one in the lecture notes for a different course, found at this link: www.profzeno.com/agreatclass/lecture10"; "I asked ChatGPT how to solve part (c)"; "I put my solution for part (c) into ChatGPT to check that it was correct and it caught a missing case.") If you use the provided LaTeX template, you can use the `sources` environment for this. Ask if you need help!

Grading

Remember, submitted homeworks are graded for completeness, not correctness. Correctness is evaluated using homework quizzes. Each submitted problem will be graded out of six points according to the following rubric:

- Does the solution address the correct problem?
- Does the solution make a reasonable attempt at solving the problem, even if not fully correct?
- Is the presentation neat?
- Is the explanation clear?

- Does the solution list collaborators or sources, or state that the student did not use any collaborators or outside resources?
- Is the solution written in the student's own voice (not copied directly from an outside resource)?

1. Suppose that you are the booking agent for breakout pop star Temple Raye. The possible gigs that you are considering can be divided into *arena* venues or *club* venues, and you have the option of booking an arena or a club every weekend. Arenas generally pay much more than clubs. However, the prep time for an arena show is longer. In particular, if you schedule Temple to play an arena, she can't play any concert (arena or club) the previous weekend. On the other hand, if you schedule her in a club, it's okay if she plays in arena or a club the weekend before.

   Given a sequence of n weekends, a *plan* is a choice of club, arena, or none for the gigs for each of the n weekends. A plan is only valid if, for any weekend where a club is chosen, none is chosen for the previous weekend. The revenue from the plan is the sum of the revenues from the chosen gigs. (Choosing none adds $0 to the revenue.) Your goal is to find an optimal plan—that is, one that is both valid and has maximal revenue.

   For weekend i, we write c_i to mean the revenue from the club show that you can book that weekend, and a_i to mean the revenue from the arena show you can book that weekend.

   Here is an example input to the problem.

	weekend1	**weekend2**	**weekend3**	**weekend4**	**weekend5**
c	10	8	2	10	24
a	11	100	40	5	50

   So, for this example, c_2 is 8 and a_4 is 5. An optimal plan for this input is to choose no show for weekend 1, the arena show for weekend 2, the club show for weekend 3, no show for weekend 4, and arena show for weekend 5, with a total revenue of $0 + a_2 + c_3 + 0 + a_5 = 0 + 100 + 2 + 0 + 50 = 152$.

   (a) Give an *invalid* plan for the example input above and explain why it is invalid.

   (b) At first, you think that a greedy algorithm that just looks one weekend ahead could work to find the optimal schedule:

   ```
   Assign i = 1
   While i ≤ n:
     If a_{i+1} > c_i + c_{i+1}:
         Choose no show for weekend i
         Choose the arena show for weekend i+1
         Set i = i+2
     Else:
         Choose the club show for weekend i
         Set i = i+1
   ```

   (To deal with problems of i going beyond n, define $c_i = a_i = 0$ when $i > n$.)

   Give an example input where the above algorithm does not correctly solve the problem.

   (c) Give an efficient (polynomial) algorithm that takes in values for $c_1, c_2, \ldots, c_n$ and $a_1, a_2, \ldots a_n$ and returns the optimal revenue (not schedule!). This will probably be easiest to do if you define a recurrence relation and write a dynamic programming algorithm based on it.

   (d) Write a recursive algorithm that can compute the optimal set of gigs to choose, given optimal total revenue values for all weekends from 1 to n.