CS 477/677 Analysis of Algorithms

Homework 6

Due April 22, 2025

1. (U & G-required) [100 points]

A manager of a construction business has to decide every day what projects to sign up for his team. The projects can be of two categories: either *repairs* (e.g. fixing a fence, replacing a window), or *construction* (e.g., building a shed, add a new patio). For a given day i, selecting a *repair* job will provide a revenue of $r_i > 0$ dollars, while for a *construction* project, it would be $c_i > 0$ dollars. However, if the company works on a *construction* project in day i, they cannot sign up for any job (of either type) in the previous day $i-1$, as they need that previous day to acquire the building materials. If the team works on a *repairs* job in day i, they can work on any job (of either type) in the previous day $i-1$.

A *plan* for the team, is specified as a choice of *repairs*, *construction* or *none*, for a sequence of n given days (with the constraint that if *construction* is selected for day $i > 1$, then *none* must be chosen for day $i-1$. It is permitted to choose a *construction* project in day 1. The **revenue of the plan** is computed as follows: for each day i, add r_i to the total if choosing *repairs* in week i, and add c_i to the total if choosing *construction* in day i (add 0 if choosing *none* in day i.)

The problem. Given a set of values $r_1, r_2, \ldots, r_n$ and $c_1, c_2, \ldots, c_n$ find a plan of **maximum** revenue. Develop a dynamic programming algorithm that finds the value of an optimal plan using the steps outlined below.

(a) [20 points] Write a recursive formula for computing the optimal value for the *total revenue* obtained by the company (i.e., define the variable that you wish to optimize and explain how a solution to computing it can be obtained from solutions to subproblems).
Submit: the recursive formula, along with definitions and explanations on what is computed (in a PDF file).

(b) [30 points] Write an algorithm that computes an optimal solution to this problem, based on the recurrence above. The algorithm should save in an output file the optimal values for

all the subproblems as well as the optimal value for the entire problem. Implement your algorithm in C/C++ and run it on the following values:

	Day 1	Day 2	Day 3	Day 4
r (repairs)	1,000	100	1,000	1,000
c (construction)	500	5,000	500	100

Submit:

- The source file containing your algorithm (name the file **`revenue_pb.c`** or **`revenue_pb.cpp`**)
- The output file created by your algorithm (name the file **`revenue_pb_out.txt`**), which contains:
  - The table with the optimal values to all subproblems (save the entire table)
  - The optimal value for the entire problem (indicate this on a separate line after the table, even if the value is found in the table above)

(c) [20 points] Update the algorithm you developed at point (b) to enable the reconstruction of the optimal solution, i.e., to **store the choices** you made when computing the optimal values for each subproblem in part (b). Your updated C/C++ program should store those choices in an auxiliary table and then save that table in an output file. Include these updates in your implementation from point (b).

Submit:

- The source file containing your algorithm (name the file **`revenue_pc.c`** or **`revenue_pc.cpp`**)
- The output file created by your algorithm (name the file **`revenue_pc_out.txt`**), which contains the values of the table containing the additional information (choices) needed to reconstruct the optimal solution (print the entire table).

(d) [30 points] Using the additional information computed at point (c), write an algorithm that prints the optimal solution, i.e., it prints what type of project the company worked on every day (*repairs*, *construction*, *none*). **Important notes**: 1) this should be a stand-alone

algorithm, separate from the ones in parts b and c; 2) the algorithm should read in the choices from the **`revenue_pc_out.txt`** file saved in part c). Implement this algorithm in C/C+.

Submit:

- The source file containing your algorithm (name the file **`revenue_pd.c`** or **`revenue_pd.cpp`**)
- The output file created by your algorithm (name the file **`revenue_pd_out.txt`**) that contains the optimal solution to the problem given by the numerical values in part (b).

2. **(G-required)** [20 points] Show how the algorithm for finding the longest common subsequence (discussed in class) operates on the following two strings:

X = < A, B, C, D, O, N, A, B>

Y = < A, C, D, O, M, A, B, C>

Give both the length and the actual solution for the longest common subsequence of X and Y.

Extra Credit

3. [20 points] Indicate whether the following statements are true or false and justify your answers.

(a) If X and Y are sequences that both begin with the character A, every longest common subsequence of X and Y begins with A.

(b) If X and Y are sequences that both end with the character A, some longest common subsequence of X and Y ends with A.