COM S 311 SPRING 2021
HOMEWORK 3

Due: April 2, 11:59 p.m.

Early submission: April 1, 11:59 p.m., (5% bonus)

Late Submission Due: April 3, 11:59 A.M. (25% penalty)
(submissions after this deadline will not be graded)

Guidelines

- For each problem, if you write the statement “I do not know how to solve this problem” (and nothing else), you will receive 20% credit for that problem. If you do write a solution, then your grade could be anywhere between 0% to 100%. To receive this 20% credit, you must explicitly state that you do not know how to solve the problem.
- You are allowed to discuss with classmates, but you must work on the homework problems on your own. You should write the final solutions alone, without consulting anyone. Your writing should demonstrate that you understand the proofs completely.
- When proofs are required, you should make them both clear and rigorous. Do not hand-waive.
- Please submit your assignment via Canvas.
  - You **must** type your solutions. Please submit a PDF version.
  - Please make sure that the file you submit is not corrupted and that its size is reasonable (e.g., roughly at most 10-11 MB).

  If we cannot open your file, your homework will not be graded.
- Any concerns about grading should be expressed within one week of returning the homework.

Problems

(1) The Elevator Problem

At a prestigious computer science department, there is an elevator that is serving n floors. The elevator has k buttons, each annotated by an integer that, when pressed, travels the integer number of floors. Note that when pressing a button that would move to a floor not served by the elevator, the elevator will not move.

For example, the elevator could serve 72 floors and have buttons annotated with integers -5 and 2. If the elevator is on floor 17 and pressing the button with integer -5 will move the elevator to floor 12. Then pressing two times the button with integer 2 will move the elevator to level 16.

It is a high sport culture for the inhabitants of this building to find out if it is possible to travel with the elevator from floor i to floor j by pressing a sequence of buttons. If so, then the inhabitants want to know a shortest such sequence and how many shortest sequences there are.

Write an efficient dynamic programming algorithm in pseudo-code that answers these questions, give a brief justification of its correctness, and analyze its runtime.

The input parameters for your algorithm should include (i) $n \in \mathbb{N}$ representing the number of floors the elevator is serving, (ii) $b_1, \ldots, b_k \in \mathbb{Z}$ ($k \in \mathbb{N}$) representing the buttons with their integer values, and (iii) $i, j \in \{1, \ldots, n\}$ where we want to know whether the elevator can travel from floor i to floor j. As output, the algorithm should state "NO" when moving from floor i to floor j is not possible. Otherwise, the algorithm should output (i) a shortest sequence of integers describing the buttons pressed to move the elevator from floor i to floor j, and (ii) the number of such sequences.

(2) The Cookie Game

Riley and Morgan play the following cookie game. Given is one set of n red cookies and another set of m green cookies. At every turn, a player must eat two cookies from one set and one cookie from the other set (i.e., either (i) two green cookies and one red cookie, or (ii) two red cookies and one green cookie). The player who cannot move loses. Assuming Riley will begin the game, which player will win?

Write an efficient dynamic programming algorithm in pseudo-code that decides whether a winning strategy for one of the players exists, give a brief justification of the correctness of your algorithm, and analyze its runtime.

Your algorithm's input parameters should include $n, m \in \mathbb{N}_0$ representing the given numbers of green and red cookies, respectively. As output, the algorithm should state whether Riley or Morgan will win, or if there is no winning strategy for either player.

(3) The Constrained LCS Problem

Let X and Y be strings over an alphabet Σ. For $\Gamma \subseteq \Sigma$ we define the Γ-*constrained longest common subsequence* of X and Y to be a longest common subsequence of X and Y that does NOT contain any of the characters in Γ.

Give a detailed proof that the problem of finding a constrained longest common subsequence exhibits optimal substructure.

(4) Minimum Spanning Trees

Let $G = (V, E)$ be a connected and undirected graph with a weight function $c\colon E \to \mathbb{R}$.

Given an arbitrary vertex $v \in V$, is it true that an edge incident to v with the least weight always belongs to some minimum spanning tree of G?

Give a thorough proof showing the correctness of your answer.