CSE Ph.D. Qualifying Exam, Fall 2023
Algorithms

This is a **closed book** exam. No books or notes are allowed.

Instructions:

Please answer three of the following four questions. All questions are graded on a scale of 10. If you answer all four, all answers will be graded and the three lowest scores will be used in computing your total.

Questions:

1. **Greedy**

   There are n houses in a village and one large water tank near by. You are required to supply water to all houses by laying pipes. For each house you can either (i) build a pipe to the water tank or (ii) build a pipe to a house that already has water. You are given array T, where *T[i]* is the cost of building a pipe between the water tank and house i. You are also given list H, where *H[i]* is a tuple *(j, c)* indicating the cost c of building a pipe between houses i and j. The cost c is non-negative.

   Your task is to supply water to all the houses at minimum cost. Describe a greedy algorithm to solve the task and prove the correctness of your algorithm. Your proof of correctness can use an exchange argument.

   (Hint: Model the problem as a graph and devise an algorithm for connecting the vertices.)

2. **Dynamic Programming**

   You are given an integer array *cost*, where *cost[i]* is the cost of the *i-th* step on a staircase. Once you pay the cost, you can either climb one step or two steps. Design a dynamic programming algorithm which finds (i) the minimum cost to reach the top floor and (ii) the sequence of steps to achieve that minimum.

   Example 1:
   Input: cost = [10, 15, 20]
   Output: minimum cost = 15, sequence of steps = [1]

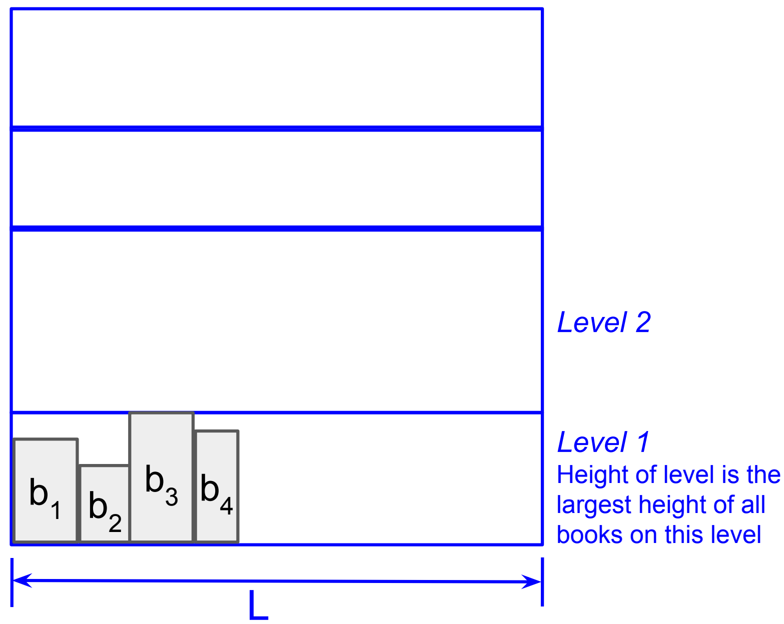
   Example 2:
   Input: cost = [1, 100, 1, 1, 1, 100, 1, 1, 100, 1]
   Output: minimum cost = 6, sequence of steps = [0, 2, 4, 6, 7, 9]

   You are required to provide the recurrence relation and write a pseudocode.

3. **Dynamic Programming: bookshelf**

   You are given n books, $b_1, b_2, ..., b_n$ that need to be arranged into a bookshelf. The books are already sorted by their indices. Each book b_i has thickness t_i and height h_i. The books must be arranged in the given order of their indices, from the lowest level to the highest level of the bookshelf. The bookshelf has a total width of L, and the height of each level on the bookshelf can be adjusted.

The aim is to minimize the *total space usage* of the n books, defined as the sum of the heights of the highest book on each level, multiplied by the bookshelf width L. An illustration is shown below (the figure may not show an optimal solution):



Example: we have three books b_1, b_2, b_3. The thickness values are: $t_1 = 1$, $t_2 = 1$ and $t_3 = 1$. The heights of the books are: $h_1 = 1$, $h_2 = 2$, $h_3 = 3$. The width of bookshelf $L = 2$. The optimal solution is to put b_1 on level 1 and put b_2 and b_3 on level 2, which results in a total space usage of 8.

Please design a dynamic programming algorithm to find the minimum total space usage of the n books. Please define the subproblem(s) and give the recurrence relations. Analyze the time and space complexity of your algorithm. Backtracing step and pseudocode are not required.

4. **NP-complete**

The 2-PARTITION problem is: given a set S of numbers, determine whether S can be partitioned into two sets, A and $S - A$, such that: $\sum_{x \in A} x = \sum_{x \in (S-A)} x$.

Please prove that the 2-PARTITION problem is NP-complete using that the SUBSET-SUM problem is NP-complete.

In SUBSET-SUM(X, k), we are given a set $X = \{x_1, .., x_n\}$ of integers and a target number k, and we want to find a subset $Y \subseteq S$ such that the members of Y sum up to k.

Hint: to construct 2-PARTITION instance S given a SUBSET-SUM instance (X, k), you can consider adding one number to X and this number can be calculated from the following two variables: (1) the sum of all numbers in X; (2) k.