Name:

Student ID #:

- You are expected to work on this assignment on your own
- Use pseudocode, Python-like or English to describe your algorithms. Absolutely no C++/C/Java
- When designing an algorithm, you are allowed to use any algorithm or data structure we explained in class, without giving its details, unless the question specifically requires that you give such details
- Always remember to analyze the time complexity of your algorithms
- Homework has to be submitted electronically on Gradescope by the deadline. No late assignments will be accepted
Problem 1. (25 points)

In the United States, coins are minted with denominations of 1, 5, 10, 25, and 50 cents. Now consider a country whose coins are minted with denominations of \( \{d_1, \ldots, d_k\} \) units. They seek an algorithm that will enable them to make change of \( n \) units using the minimum number of coins.

1. The greedy algorithm for making change repeatedly uses the biggest coin smaller than the amount to be changed until it is zero. Provide a greedy algorithm for making change of \( n \) units using US denominations. Prove its correctness and analyze its time complexity.

2. Show that the greedy algorithm does not always give the minimum number of coins in a country whose denominations are \( \{1, 6, 10\} \).

3. Give dynamic programming algorithm that correctly determines the minimum number of coins needed to make change of \( n \) units using denominations \( \{d_1, \ldots, d_k\} \). Analyze its running time.

Answer:
Problem 2. (25 points)

Given an array $A = \{a_1, a_2, \ldots, a_n\}$ of integers, we say that a subsequence $\{a_{i_1}, a_{i_2}, \ldots, a_{i_t}\}$ is (monotonically) increasing if for every $i_s < i_t$, we have $a_{i_s} < a_{i_t}$. Given an array $A$ of size $n$, we want to compute the length of the longest increasing subsequence (LIS) in $A$. For instance, if $A = \{9, 5, 2, 8, 7, 3, 1, 6, 4\}$ the length of the LIS is 3, because $(2, 3, 4)$ (or $(2, 3, 6)$) are LIS of $A$. Give a $O(n^2)$ dynamic programming algorithm for this problem. Analyze the time- and space-complexity of your solution.

Answer:
Problem 3. (25 points)

You have a set of $n$ jobs to process on a machine. Each job $j$ has a processing time $t_j$, a profit $p_j$ and a deadline $d_j$. The machine can process only one job at a time, and job $j$ must run uninterruptedly for $t_j$ consecutive units of time. If job $j$ is completed by its deadline $d_j$, you receive a profit $p_j$, otherwise a profit of 0. You can assume that all parameters are integers, and that the jobs are sorted in increasing order of deadline. Give a dynamic programming algorithm to the problem of determining the schedule that gives the maximum amount of profit. Analyze the time- and space-complexity of your solution.

Answer:
Problem 4. (25 points)

We are given a list of $n$ items with sizes $s_1, s_2, \ldots, s_n$. A sequential bin packing of these items is an assignment of items to bins, such that in each bins the items are consecutive. (That is, each bin has items $s_i, s_{i+1}, \ldots, s_j$ for some indices $i < j$.) Bins have unbounded capacities. The load of a bin is the sum of the elements in it. Give an algorithm that determines a sequential packing of $n$ items into $k$ bins for which the maximum load of a bin is minimized. Analyze the time-complexity and space-complexity.

Answer: