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)

Let \( A = \{a_1, a_2, \ldots, a_n\} \) be a set of \( n \) positive integers and let \( T \) be another integer. Design a dynamic programming algorithm that determines whether there exists a subset of \( A \) whose total sum is exactly \( T \). Analyze the time- and space-complexity of your solution. For instance, if \( A = \{4, 5, 17, 23, 11, 2\} \) and \( T = 35 \) the algorithm should return \text{TRUE} because the subset \( \{5, 17, 11, 2\} \) sums to 35. For the same set of numbers if we choose \( T = 31 \) the problem has no solution, and the algorithm will return \text{FALSE}.

Answer:
Problem 2. (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 3. (25 points)

Let $A$ be a $n \times m$ matrix of 0’s and 1’s. Design a dynamic programming $O(nm)$ time algorithm for finding the largest square block of $A$ that contains 1’s only.

**Hint:** Define the dynamic programming table $l(i,j)$ be the length of the side of the largest square block of 1’s whose bottom right corner is $A[i,j]$.

**Answer:**
Problem 4. (25 points)

A string \( y \) is a palindrome if \( y^R = y \), where \( y^R \) is the reverse of \( y \). Given a text \( x \) a partitioning of \( x \) is a palindrome partitioning if every substring of the partition is a palindrome. For example, \( abab|b|aba|bb|a|b|aba \) and \( abal|b|bbabb|ababa \) are two palindrome partitioning of \( x = ababbbabbababa \). Design a dynamic programming algorithm to determine the coarsest (i.e., fewest cuts) palindrome partitioning of \( x \). In the example, the second partition (3 cuts) is optimal. Analyze the time- and space-complexity of your solution.

Answer: