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. (10 points)

Let $G = (V, E)$ be a weighted directed graph with weight function $w : E \rightarrow \{0, 1, \ldots, W\}$ for some nonnegative integer $W$. Modify Dijkstra’s algorithm to compute the shortest paths from a given vertex $s$ in $O((n + m) \log W)$-time.

Answer:
Problem 2. (10 points)
Suppose that we are given a set of \( n \) objects (initially each item in its own set) and we perform a sequence of \( m \) Make-set, Find-set, and Link operations, where all the Link operations occur before any of the Find-set operations. Show that the resulting sequence will take only \( O(m) \) time (if both path compression and union by rank are used).

What happens in the same situation if only the path-compression heuristic is used?

**Hint:** Use the accounting method. The key observation is that since all the Find-sets are done after the Links, once a node \( x \) appears on a find path, \( x \) will be either the root or a child of a root at all times thereafter.

**Answer:**
Problem 3. (10 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. We want to devise an algorithm that will enable us 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 an efficient 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: