Fundamental Algorithms, Section 003
Homework 10, Fall 22. Due date: Wednesday, November 30, 4:55pm on Gradescope

1. Suppose you are given a directed graph $G = (V, E)$, and an array $\mathrm{Rwd}[1 : n]$ of non-negative integers, where $\mathrm{Rwd}[v]$ is a reward associated with vertex v. You are also given a start vertex s and a destination vertex t. Give an algorithm to calculate the maximum reward on any path from s to t, including paths with cycles, where you collect the reward from each vertex once, however often it is visited. Explain why your solution produces the correct result. Also, justify your running time.
Hint: What should you do if G is strongly connected?

2. Suppose you are given an undirected graph $G = (V, E)$, and a designated vertex $s \in V$. We define the length of a path in G to be the number of edges on the path. Give a linear time algorithm to determine for each vertex $v \in V$ the number of shortest paths from s to v.
Hint: Augment a Breadth First Search from s, and explain why your augmentation produces the correct result. Also, justify your running time.

3. Let $G = (V, E)$ be a directed graph, where all edges have positive lengths, and let s be a designated source vertex. Suppose you want to find shortest paths from s to each vertex $v \in V$ and in the event of ties, among these shortest length paths find one with fewest edges. Explain how to modify Dijkstra's algorithm and the path reconstruction, while still achieving the same runtime as Dijkstra's algorithm. Justify your running times and explain why your modifications produce the correct result.

4. Suppose you are given a directed graph $G = (V, E)$, where all edges have positive lengths, and you are also give a designated vertex $s \in V$. It may be that there is more that one shortest path from s to any given $v \in V$.

By modifying Dijkstra's algorithm and the path recovery procedure, give an algorithm that constructs the directed acyclic graph of shortest paths from s to every other vertex. The modified Dijkstra's algorithm should still run in $O((n+m)\log n)$ time and the modified path recovery procedure should run in $O(n + m)$ time. Justify your running times and explain why your modifications produce the correct result.

Challenge problems. Do not submit.
5. Consider the single source shortest path problem, with non-negative integer edge lengths, and with the additional guarantee that every shortest path has length at most ℓ. Implement Dijkstra's algorithm so that it runs in time $O(n + m + \ell)$.