# CSE 101 Homework 2

Winter 2021

This homework is due on gradescope Friday January 29th at 11:59pm pacific time. Remember to justify your work even if the problem does not explicitly say so. Writing your solutions in LaTeXis recommend though not required.

**Question 1** (Marble Run, 35 points)**.** *Connie is building a marble run. She has built a complicated downhill, branching track for marbles to run along represented by a DAG $G$. When a marble hits a vertex of this graph that it not a leaf, it will travel along a random one of the outgoing edges. With some experimentation, Connie has computed for each vertex the probability that the marble will leave through each of its outgoing edges.*

*Give an algorithm that given $G$, these probabilities and the vertex $s$ where the marble starts, computes for each vertex $v$ of $G$, the probability $p(v)$ that a marble running through this course will at some point pass through $v$. For full credit, your algorithm should run in linear time. Hint: Compute the values of $p(v)$ one at a time in an appropriate order.*

**Question 2** (Contracting Cycles, 20 points)**.** *Consider the following method of computing the metagraph of a graph $G$. If $G$ is a DAG, return $G$. Otherwise, find some cycle $C$ in $G$ and let $G'$ be the graph obtained from $G$ by replacing all of the vertices of $C$ with a single vertex with edges to/from the same vertices that have edges from/to some vertex of $C$. Then recursively compute the metagraph of $G'$. Prove that this algorithm correctly computes the metagraph.*

**Question 3** (Graph Cycle, 15 points)**.** *Is it the case that for every finite, strongly connected directed graph $G$ that there is a cycle that visits each vertex of $G$ at least once, but uses no edge more than once? Prove or provide a counter-example.*

**Question 4** (Dijkstra at Small Distances, 30 points)**.** *Suppose that you are given a graph $G$ with positive integer edge weights, a vertex $s$ and an integer $L$. Give an algorithm that determines which other vertices $w$ in $G$ have paths from $s$ to $w$ of length at most $L$. For full credit your algorithm should run in time $O(|V| + |E| + L)$. Hint: You will want to devise some appropriate modification of Dijkstra's algorithm that takes advantage of the fact that you only need to keep track of distances that are less than $L$.*

**Question 5** (Extra credit, 1 point)**.** *Approximately how much time did you spend working on this homework?*