CSE 40113: Algorithms
Homework 6

You may complete this homework in groups of 3 or less students. Note that the integrity policy applies: your group should write up your own work, although you're welcome to work on the problems in a larger group. If you have any questions, please re-read both the homework guidelines and the academic integrity policy carefully, and then come discuss any questions or concerns with me.

Required Problems

1. For this problem, you're doing to put yourself in the mindset of a network engineer, working for a company that is facing a routing question. You have a connected graph $G = (V, E)$, where the nodes in V represent sites that wish to communicate with each other. Each edge e is weighted with a *bandwidth* $b(e)$, which is the maximum amount of information it can transmit.

   For every pair of nodes, the goal is to select a single path between the nodes along which they can communicate. However, the rate of transfer is limited by the smallest edge on that path, which we sometimes called the bottleneck edge: so for a path P, the bandwidth of the path is equal to $\min_{e \in P} b(e)$. (If there are no paths from s to t, then the bandwidth is $-\infty$.)

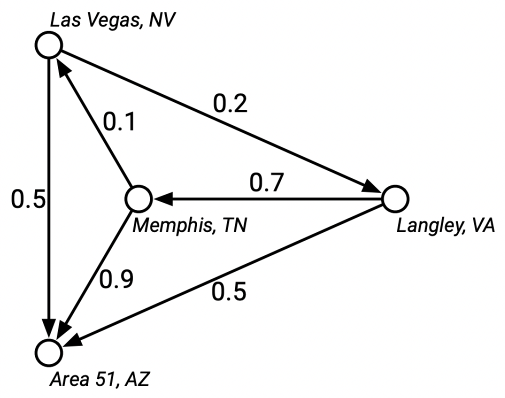
   One of the network designers has suggested that instead of storing a path for each pair of vertices, you could instead store a single spanning tree such that the unique path between vertices in the tree in fact achieves the largest possible bandwidth of any path. While skeptical, after some attempts to prove your colleague incorrect, you must admit that this may in fact be true.

   (a) Prove that there is a spanning tree T such that, for every pair of vertices s and t in the graph, the tree contains the maximum bandwidth path between s and t, and give an algorithm to compute it.

   (b) Describe an algorithm to solve the following problem, as efficiently as possible: Given an undirected weighted graph G, two vertices s and t, and a weight W, is the bottleneck distance between s and t at most W?

2. Mulder and Scully have computed, for every road in the United States, the exact probability that someone driving on that road won't be abducted by aliens. Agent Mulder needs to drive from Langley, Virginia to Area , Nevada. What route should he take so that he has the least chance of being abducted?

   More formally, you are given a directed graph $G = (V, E)$, where every edge e has an independent safety probability $p(e)$. The safety of a path is the product of the safety probabilities of its edges. Design and analyze an algorithm to determine the safest path from a given start vertex s to a given target vertex t. You may assume that all necessary arithmetic operations can be performed in $O(1)$ time.



For example, with the probabilities shown above, if Mulder tries to drive directly from Langley to Area 51, he has a 50% chance of getting there without being abducted. If he stops in Memphis, he has a $0.7 \times 0.9 = 63\%$ chance of arriving safely. If he stops first in Memphis and then in Las Vegas, he has a $1 - 0.7 \times 0.1 \times 0.5 = 96.5\%$ chance of being abducted! (That's how they got Elvis, you know.)

3. After a grueling midterm, you are taking the bus home. Since you planned ahead, you have a schedule that lists the times and locations of every stop of every bus in South Bend. Unfortunately, no single bus will get you home, so you must change buses at least once. There are exactly b different buses. Each one starts running at 12:01am, makes exactly n stops throughout the day, and stops running at 11:50pm. Buses run exactly on schedule in this theoretical version of South Bend, and you have a perfectly on time watch to plan with. Finally, you are exhausted, so you don't want to walk between bus stops at all.

   (a) Describe and analyze an algorithm to determine the sequence of bus rides that get you home as *early* as possible. You goal is to minimize arrival time, NOT time spent traveling.

   (b) Things just got stranger! There are now zombies infesting the city. The transit authority doesn't have the funding to zombie-proof the bus stops (although you're safe on a bus), so your goal just changed. Describe and analyze an algorithm to determine a sequence of bus rides that minimizes the *total time you spend waiting at bus stops*; you don't care how long you spend on a bus, or what time you get home. (Assume you can wait inside the building for that first bus to arrive, so you're safe for that waiting period - it's all the bus stops that are risky.)

   Hint: for this problem, not that I am NOT giving you a graph. So, if you're using any graph algorithm, you'll have to build the (presumably weighted) graph before you can use any of the algorithms we have covered! In other words, you're doing a reduction here, at least if you're using a graph algorithm (which I highly recommend).