Assignment 5

Due: February 25, 2019

Your solutions must be typed (preferably typeset in \LaTeX) and submitted as a hard-copy at the beginning of class on the day its due.

When asked to provide an algorithm you need to give well formatted pseudocode, a description of how your code solves the problem, and a brief argument of its correctness.

Problem 1: Highway Safety  [10 points]  As is well-known, America's highway infrastructure is crumbling. Yet travel must continue. Suppose you are given a map of U.S. cities and roads connecting them that shows, for every road segment, the probability of traveling down that segment safely, i.e. without destroying your axle in a pothole, falling into a river due to a broken bridge, etc. Design and analyze an algorithm to determine the safest route from Portland to your preferred summer vacation spot.

Stated more formally, suppose 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 $t$.

Problem 2: MST  The cut property makes it possible to build minimum spanning trees greedily, starting from an empty graph and adding one edge at a time. A different approach is to start with the original graph and remove edges greedily, one at a time, until an MST remains. A scheme of this second type can be justified by the following property.

Pick any cycle in the graph, and let $e$ be the heaviest edge in that cycle. Then there is a minimum spanning tree that does not contain $e$.

(a) [5 points]  Prove this cycle property.

(b) [5 points]  Use the property to justify the following MST algorithm. The input is an undirected graph $G = (V, E)$ with edge weights $\{w_e\}$.

1. sort the edges according to their weights
2. for each edge $e \in E$, in decreasing order of $w_e$
3. \hspace{1cm} if $e$ is part of a cycle of $G$
4. \hspace{3cm} $G = G - e$ (that is, remove $e$ from $G$)
5. return $G$
(c) [5 points] On each iteration, the algorithm must check whether there is a cycle containing a specific edge $e$. Give a linear-time algorithm for this task, and justify its correctness.

(d) [5 points] What is the overall time complexity of this algorithm, in terms of $|E|$? Explain your answer.