ESO207A: Data Structures and Algorithms

Homework *4a: Shortest Paths* Due Date: 6 November 2017

Instructions.

1. Start each problem on a new sheet. For each problem, Write your name, Roll No., the problem number, the date and the names of any students with whom you collaborated.
2. For questions in which algorithms are asked for, your answer should take the form of a short write-up. The first paragraph should summarize the problem you are solving and what your results are (time/space complexity as appropriate). The body of the essay should provide the following:
   (a) A clear description of the algorithm in English and/or pseudo-code, where, helpful.
   (b) At least one worked example or diagram to show more precisely how your algorithm works.
   (c) A proof/argument of the correctness of the algorithm.
   (d) An analysis of the running time of the algorithm.

   Remember, your goal is to communicate. *Full marks will be given only to correct solutions which are described clearly.* Convoluted and unclear descriptions will receive *low marks.*

Problem 1. *CLRS 24.3-6* Given a directed weighted graph $G = (V, E)$, where each edge $(u, v) \in E$ has an associated value $r(u, v)$ which is a real number $0 \le r(u, v) \le 1$ that represents the *reliability* of the communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail and we assume that these probabilities are independent. Give an $O((|V| + |E|) \log(|V|)$ time algorithm to find the most reliable path from a given source vertex to all the other vertices in V. Argue the modelling of your problem carefully and hence the complexity of your algorithm. (You do not have to restate or redo any algorithm already done in the class).

Problem 2. Let $G = (V, E)$ be a given weighted, directed graph with non-negative edge weights. You are given a table of claimed shortest path distances $u.d$ and predecessor $u.\pi$ for each $u \in V$. Can you check in $O(|V| + |E|)$ time whether the d and π values correspond to some shortest path tree? Is the condition of non-negative edge weights really necessary?

Problem 3. *CLRS 24.1-4* Modify the Bellman-Ford algorithm so that it sets $v.d$ to $-\infty$ for all vertices v for which there is a negative-weight cycle on some path from the source to v.

Problem 4. *CLRS 24.1-6* Given a weighted, directed graph $G = (V, E)$ with a negative weight cycle. Give an efficient algorithm ($O(|V|\,|E|)$ time) to list the vertices of *one such cycle.* Prove that your algorithm is correct.