

HOMEWORK 7

Due Friday, November 6, 2009 at the beginning of class

For general homework policies and our suggestions, please see the policy document. Do not forget to look at Jeff's grading rubric before you write down your solutions. No collaboration is allowed on problem 1.

1. (40 points) (**Your must work on this problem on your own: NO collaboration is allowed**)

In class I mentioned that Dijkstra's algorithm does not work when the edge lengths can be negative. Give an example that exhibits this. That is, come up with an example of a directed graph along with edge lengths (one of which is negative) and two nodes s and t in the graph such that when Dijkstra's algorithm is run on the graph, it does not output the shortest path between s and t . (As before, the length of a path is the sum of the lengths of the edges in it.)

Note: You have the full freedom in picking the graph.

2. (40 points) In this problem, you will tackle the so called the *bottleneck path problem*. Like the shortest path problem, in this problem you are also given a directed graph $G = (V, E)$ and for every edge $e \in E$, we are given its length $\ell_e \geq 0$. The problem differs from the shortest path problem in its objective. For the bottleneck path problem, given two nodes $s, t \in V$, you need to find a path with the shortest longest edge. More precisely, given a path P connecting s and t in G , its bottleneck is defined as $b(P) = \max_{e \in P} \ell_e$. The goal is to output an $s - t$ path P^* that minimizes $b(P)$ over all $s - t$ paths P .

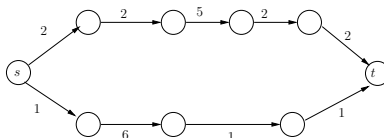


Figure 1: In the graph above, the bottom path is the shortest path (with length 9) while the top path has the minimum bottleneck (of 5).

Present an algorithm that solves the problem above and prove its correctness. (Justify why your algorithm runs in polynomial time: a detailed analysis of its run time is not required.)

Hint: Try to modify Dijkstra's algorithm.

3. (20 points) Exercise 6 in Chapter 4.

Note: As mentioned in the previous homework, in many real life problems, not all parameters are equally important. Also sometimes it might make sense to "combine" two parameters into one. Keep these in mind when tackling this problem.

Hint: In the solution that I have in mind, the analysis of the algorithm's correctness follows the exchange argument.