McGill University COMP360 Winter 2011 **Instructor: Phuong Nguyen**

Assignment 10

Due April 6 before the lecture

The work you submit must be your own. You may discuss problems with each others; however, you should prepare written solutions alone. Copying assignments is a serious academic offense, and will be dealt with accordingly.

Question 1 (5pt) Recall the Bellman-Ford algorithm for the (general) Shortest Path problem. In this question you are asked to write a program that computes the total number of shortest st-paths in a given graph.

Formally, the input to your algorithm consists of a directed graph G and two vertices s and t in G. Each edge e of G is associated with a cost $c(e)$ that may be negative; however there is no cycle in G that has negative total cost. Your algorithm must output the total number of st-paths in G of minimum total cost. Note that you do not have to compute these paths.

Question 2 (10pt) (a) [**5pt**] Consider a directed grid graph G whose vertices are point (i, j) on the plane, for integers i, j: $0 \le i \le m$ and $0 \le j \le n$. The edges in G are horizontal and vertical grid edges that go from left to right and from bottom to top, together with diagonal edges in the direction from the lower-left corner $(0, 0)$ to the upper-right corner (m, n). In other words, the edges are:

$$\begin{aligned} ((i,j),(i,j+1)) &\qquad \text{for } 0 \le i \le m, 0 \le j \le n-1 \\ ((i,j),(i+1,j)) &\qquad \text{for } 0 \le i \le m-1, 0 \le j \le n \\ ((i,j),(i+1,j+1)) &\qquad \text{for } 0 \le i \le m-1, 0 \le j \le n-1 \end{aligned}$$

Each edge e of G is associated with a cost $c(e)$ which is a non-negative integer.

Given a path P in G from $(0, 0)$ to (m, n). Show how to modify the costs on the edges of G so that P is the unique minimum-cost path in G if and only if it is a minimum-cost path under the new cost function.

(b) [**5pt**] Give an algorithm that runs in time $\mathcal{O}(mn)$ and space $\mathcal{O}(m+n)$ that determines whether G has a unique minimum-cost path from $(0, 0)$ to (m, n). Justify the time and space complexity of your algorithm. Use (a) to argue that your algorithm is correct.

Question 3 (10pt) Consider the following problem. There are m machines $M_1, M_2, \ldots, M_m$. There are k types of job, and there are n jobs in total. (In general $n \ge k$, so there can be multiple jobs of the same type.) Each machine M_i is capable of processing a set of types of jobs, denoted by S_i. For example, if $S_2 = \{5, 9, 12\}$ then machine M_2 can process jobs of types 5, 9 and 12. Assume that each job requires one unit of time and must be processed by a single machine that is capable of processing it. Furthermore, each machine M_i has a total t_i units of time available. The problem is to schedule, whenever possible, all jobs on the machines in such a way that meet the described specification. Set up a flow network for solving this problem.

(a) Clearly specify the vertices, the edges, and the capacity on each edge of the network. Specify an algorithm for computing a maximum flow of the network.

(b) Give an algorithm that determines whether it is possible to schedule all jobs in such a way that satisfies the specification above, and if so, outputs such a schedule. (The output should be a list L_i for each machine M_i; this is the list of jobs that will be processed by the machine.)

(c) Prove that your algorithm in (b) is correct.