CS 357 Assignment #4

Greg Plaxton

October 12, 2011

There are two parts to this assignment, which are described in Sections 1 and 2, respectively.

1 Programming & Problem Solving

This part of the assignment consists of a programming task and a collection of related exercises. The programming task is due by 8pm on Wednesday, October 19, and the solutions to the related exercises are due at the start of class on Wednesday, October 19. Students are allowed to work on this part of the assignment with a partner, and are strongly encouraged to do so. Each team should turn in only one program, and only one set of solutions to the related exercises. If you are having trouble finding a partner, send me an email and I will try to match you up with someone else in the same situation.

1.1 Exercises

1. A CBG $G = (U, V)$ is *ping-feasible* if it admits a matching M such that $|M| = |U|$. Consider the following computational task. The input consists of a ping-feasible CBG G, the stable matching of G, and a ping u that does not belong to U. (As established in Assignment 2, any CBG has a unique stable matching.) The goal is to determine whether the CBG $G' = (U + u, V)$ admits a matching of cardinality $|U| + 1$, and if so, to compute the stable matching of G'. Describe a polynomial time algorithm to perform this task.

2. A CBG $G = (U, V)$ is *ping-weighted* if $weight(v) = 0$ for all pongs v in V. Use the framework of the matroid greedy algorithm (described in Assignment 3), together with the algorithm that you developed in question 1 above, to obtain a polynomial time algorithm for computing an MWMCM of a ping-weighted CBG $G = (U, V)$.

1.2 Programming Task

Your program will read input from standard input, and write output to standard output. The first line of the input contains a nonnegative integer k that specifies the number of instances to follow. The integer k is followed by k "input blocks". Your program will produce k "output blocks", one for each input block. Each input block specifies a ping-weighted CBG G in exactly the same manner as in Assignment 1. The corresponding output block consists of a single line specifying a particular MWMCM of G, as specified below. The output matching should be printed in the same format as we used to print out each MWMCM in Assignment 1.

A ping-weighted CBG can have many MWMCMs. For ease of grading, you are asked to produce as output a specific MWMCM that we now describe. Let the input CBG G be (U, V), let k denote the cardinality of an MCM of G, and let $\mathcal{U}$ denote the set of all subsets U_0 of U such that U_0 is the set of pings matched by some MWMCM of G. Thus each set in $\mathcal{U}$ has cardinality k. We define a total order over the sets in $\mathcal{U}$ as follows. Let U_0 and U_1 be distinct sets in $\mathcal{U}$. Let α_0 (resp., α_1) be the k-tuple consisting of the pings in U_0 (resp., U_1), arranged in nonincreasing order of weight, with ties broken in favor of the lower ping (i.e., using the total order over pings defined in Assignment 1). Then the inequality $U_0 < U_1$ holds if α_0 lexicographically precedes α_1. Let U' denote the minimum set in $\mathcal{U}$ with respect to the total order just defined. Let G' denote the CBG (U', V). Then your program should produce as output the stable matching of G', which is guaranteed to match all of the pings in U', and hence is an MWMCM of G.

2 Textbook Exercises

This part of the assignment is due at the beginning of class on Wednesday, October 26.

1. Problem 6.6, page 317. Hint: This problem is similar to the segmented least squares problem discussed in Section 6.3.

2. Problem 6.12, page 323.

3. This question is a modified version of Problem 6.28, page 334. For parts (a) and (b), consider instead a weighted version of the problem in which each job has a positive integer weight, and the goal is to determine a maximum-weight schedulable set. Here we add a third part to the question, part (c): Give an $O(n^2)$-time dynamic programming algorithm for the original unweighted problem.

4. Problem 7.11, page 420.