CS 357 Assignment #5

Greg Plaxton

October 28, 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 Monday, November 7, and the solutions to the related exercises are due at the start of class on Monday, November 7. 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 A Deletion-Based Variant of the Matroid Greedy Algorithm

As discussed in the Assignment 3 description, Kruskal's algorithm is based on the framework of the matroid greedy algorithm. Recall that Kruskal's algorithm processes the edges in nondecreasing order of weight, and when edge e is processed, it is added to the output set if and only if it does not create a cycle with previously added edges. As discussed on page 143 of the textbook (see the last bullet on that page, which continues on page 144), another way to compute an MST is to start with all of the edges in the output set, and then to process the edges in nonincreasing order of weight: When edge e is processed, it is removed from the output set if and only if the resulting graph remains connected.

This variation of the matroid greedy algorithm works for all matroids, and not only the graphic matroid. In general, we can find a minimum weight maximal independent set of a matroid $(S, \mathcal{I})$ by initializing the output set to S and then processing the elements of S in nonincreasing order of weight: To process element x, we remove x from the output set if and only if the resulting set contains a maximal independent set. The same approach can be used to compute a maximum weight maximal independent set; we simply sort the set S in nondecreasing order, instead of nonincreasing order.

1.2 Exercises

1. Consider the following computational task. The input consists of a CBG $G = (U, V)$, the stable matching M of G, and a pong v that belongs to V. (As established in Assignment 2, any CBG has a unique stable matching.) The goal is to determine whether the CBG $G' =$

$(U, V - v)$ admits a matching of cardinality $|M|$, 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 *pong-weighted* if $weight(u) = 0$ for all pings u in U. Use the framework of the deletion-based variant of the matroid greedy algorithm described in Section 1.1, together with the algorithm that you developed in question 1 above, to obtain a polynomial time algorithm for computing an MWMCM of a pong-weighted CBG $G = (U, V)$. Remark: At the outset, use the stable matching algorithm of Assignment 2 to compute the stable matching of G.

1.3 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 pong-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 pong-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 $\mathcal{V}$ denote the set of all subsets V_0 of V such that V_0 is the set of pongs matched by some MWMCM of G. Thus each set in $\mathcal{V}$ has cardinality k. We define a total order over the sets in $\mathcal{V}$ as follows. Let V_0 and V_1 be distinct sets in $\mathcal{V}$. Let α_0 (resp., α_1) be the k-tuple consisting of the pongs in V_0 (resp., V_1), arranged in nondecreasing order of weight, with ties broken in favor of the lower pong (i.e., using the total order over pongs defined in Assignment 1). Then the inequality $V_0 < V_1$ holds if α_0 lexicographically precedes α_1. Let V' denote the maximum set in $\mathcal{V}$ 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 pongs in V', and hence is an MWMCM of G.

2 Textbook Exercises

This part of the assignment is due at the beginning of class on Monday, November 14.

1. Problem 7.10, page 419.
2. Problem 7.12, page 420.
3. Problem 7.22, page 428.
4. Problem 8.3, page 505.