CS 331 Assignment #2

Greg Plaxton

January 29, 2014

NOTE ADDED 1/31/14. The original version of this assignment contained a typo in the proof of Lemma 1. The graph G' was defined as $(U, V, M \oplus M')$, but should have been defined as $(U + u^*, V, M \oplus M')$. END OF NOTE.

There are three parts to this assignment. The first part, described in Section 1.1 below, consists of four paper-and-pencil exercises. You are required to turn in solutions to any two of these four exercises. (If you turn in more than two solutions, the grader will arbitrarily choose two of them to grade.) You are encouraged to attend the discussion sections and office hours to get hints on how to solve these exercises. You are also permitted to ask for clarifications on Piazza. You are not allowed to work with other students on these exercises. This part is due at the start of class on Wednesday, February 5. Please note that no extension will be given on this part of the assignment!

The second part, described in Section 1.2 below, consists of a programming task that is due by 8pm on Monday, February 10. As in Assignment 1, you should follow the instructions in the document entitled "Guidelines for Programming Tasks", which may be found in the *Assignments* section of the class website. For this programming assignment, the secondary deadline is 8pm on Monday, February 17.

The third part, described in Section 2 below, consists of recommended exercises. There is nothing to be turned in for this part of the assignment, as these exercises will not be graded. You are encouraged to work on the recommended exercises in order to prepare for the tests.

1 Programming & Problem Solving

A *matching* of a (undirected) graph $G = (V, E)$ is a subset M of E such that exactly $2|M|$ vertices in V are incident on (i.e., an endpoint of) some edge of M. These $2|M|$ vertices are said to be "matched" in M; the remaining vertices are said to be "unmatched" in M.

A *maximum-cardinalty matching (MCM)* of a graph G is a matching M of G such that $|M| \geq |M'|$ for all matchings M' of G.

The *weight* of a matching M of an edge-weighted graph is defined as the sum of the weights of the edges in M. A classic problem in combinatorial optimization is to determine a maximum-weight matching of a given edge-weighted graph.

A *maximum-weight MCM (MWMCM)* of an edge-weighted or vertex-weighted graph G is an MCM of G with weight at least as high as that of any other MCM of G.

A graph $G = (V, E)$ is *bipartite* if the set of vertices V can be partitioned into two sets V_0 and V_1 such that every edge in E has one endpoint in V_0 and one endpoint in V_1. Bipartite graphs arise in numerous practical applications, and many efficient algorithms have been developed that exploit their special structure.

In most applications involving bipartite graphs, the bipartition of the vertices is explicitly specified as part of the input. In such cases, we generally prefer to view a bipartite graph as a triple (U, V, E) where U denotes the vertices on the "left" side, V denote the vertices on the "right" side, and every edge in E has one endpoint in U and one endpoint in V.

The following lemma will turn out to be useful for us. The proof given below is incomplete. Exercises 1 and 2 in Section 1.1 ask you to fill in the missing parts of the proof.

Lemma 1. *Let $G = (U, V, E)$ be an edge-weighted bipartite graph, let M be an MWMCM of G, and let the edge-weighted bipartite graph $G^* = (U + u^*, V, E \cup E^*)$ be obtained from G by introducing a new vertex u^* and a set of edges E^* incident on u^*, where each edge in E^* connects u^* to some vertex in V. Then there is an MWMCM M^* of G^* such that any vertex in U that is unmatched in M is also unmatched in M^*.*

Proof. Let $\mathcal{M}$ denote the set of all MWMCMs of G^*. Let M' denote an MWMCM in $\mathcal{M}$ that maximizes $|M \cap M'|$. (In other words, for all MWMCMs M'' in $\mathcal{M}$, we have $|M \cap M''| \leq |M \cap M'|$.) Consider the bipartite graph $G' = (U + u^*, V, M \oplus M')$. (The notation $M \oplus M'$ denotes the symmetric difference of the edge sets M and M', i.e., $(M \setminus M') \cup (M' \setminus M)$.) Since M and M' are matchings, every vertex in G' has degree at most two. It follows that each connected component of G' is either a path or a cycle.

Suppose that the claim of the lemma is false. Then there is a vertex u in U that is unmatched in M and matched in M'. Since u has degree 1 in G', we deduce that the connected component of G' that contains u is a path of positive length, call it P, and that u is one of the two endpoints of P. Let A denote the set of edges in P that belong to M, and let A' denote the set of edges in P that belong to M'. Observe that the edges of P alternate between A and A'.

We complete the proof by deriving a contradiction in each of the following two cases.

Case 1: Path P is of odd length (i.e., P has an odd number of edges). See Exercise 1 in Section 1.1.

Case 2: Path P is of even length. See Exercise 2 in Section 1.1. $\square$

1.1 Paper-and-Pencil Exercises

As indicated at the beginning of this handout, you are only required to turn in solutions to two of the four exercises in this section.

Exercise 1. Complete the missing part of the proof of Lemma 1 corresponding to Case 1. Hint: Begin by arguing that u^* does not belong to P.

Exercise 2. Complete the missing part of the proof of Lemma 1 corresponding to Case 2. Hint: As in Exercise 1, begin by arguing that u^* does not belong to P.

The maximum-weight assignment problem that we addressed in Assignment 1 can be viewed as a special case of the problem of computing an MWMCM of an edge-weighted bipartite graph $G = (U, V, E)$. To see this, we can think of the set V as containing a vertex for each item. We can think of U as containing a vertex for each bid, and a vertex for each item. (The item vertices in U are used to model the reserve prices; see below.) A vertex u in U corresponding to a linear bid with intercept a and slope b has an edge to each item v in V; the weight of each such edge (u, v) is $a + b \cdot q$ where q denotes the quality of item v. A vertex u in U corresponding to a single-item bid with offer a on item v has a single adjacent edge (u, v) with weight a. Likewise a vertex u in U corresponding to an item v with reserve price a has a single adjacent edge (u, v) with weight a. (Thus a vertex u in U corresponding to an item v acts like a single-item bid on item v; in effect, we are introducing a dummy bid to model the reserve price.) It is not hard to see that we can solve an instance of the maximum-weight assignment problem of Assignment 1 by computing an MWMCM of the edge-weighted bipartite graph G just described. Remark: Due to the presence of the $|V|$ reserve price vertices in U, an MCM of G has size $|V|$.

In the following we refer to a bipartite graph $G = (U, V, E)$ arising from an instance of the maximum-weight assignment problem of Assignment 1 as a *configuration*.

For any configuration $G = (U, V, E)$ and any subset U' of U, let us define the predicate $Q(G, U')$ to hold if there exists an MWMCM M of G such that the set of vertices in U that are matched in M is equal to U'.

Exercise 3. Given a configuration $G = (U, V, E)$ and a subset U' of U such that $Q(G, U')$ holds, explain how to compute an MWMCM of G in $O(|V| \log |V|)$ time. Hint: Make use of sorting.

Exercise 4. Let $G = (U, V, E)$ be a configuration and let G' be a configuration obtained from G by adding a new bid u^*. Given G' and an MWMCM M of G, explain how to compute an MWMCM of G' in $O(|V|^2)$ time. Hint: Make use of Lemma 1 and the result of Exercise 3.

1.2 Programming Task

Your programming task is to implement the algorithm associated with Exercise 4 to solve the same computational problem as in Assignment 1. Your program should work "incrementally", in the following sense: As each successive bid is read in, your program should compute an MWMCM of the configuration corresponding to the prefix of the bids thus far revealed (and all of the items). Each successive bid should be processed in $O(n^2)$ time, where n denotes the number of items.

Within a few days we will provide sample input and output files that you can use to test your program. See the *Assignments* section of the class website to obtain these files. Any updates or clarifications to the assignment will also be posted there.

2 Recommended Exercises

1. Problem 3.10, page 110.
2. Problem 3.12, page 112.
3. Problem 4.15, page 196.
4. Problem 4.28, page 203.