

**Due on September 26, 2013 before lecture**

**Problem 1**

Given a unit-capacity directed graph with  $n$  vertices and  $m$  edges, show that  $O(n^{2/3})$  blocking flows suffice in finding a maximum flow. (Recall that we proved an upper bound of  $O(n)$  blocking flows for arbitrary graphs in class.) Now, use this property to obtain an  $O(mn^{2/3})$ -time maximum flow algorithm for unit-capacity directed graphs.

**Problem 2**

(a) Given a directed graph with  $n$  vertices and  $m$  edges, where every vertex has either in-degree or out-degree at most 1, show that  $O(n^{1/2})$  blocking flows suffice in finding a maximum flow.

(b) A *matching* in an undirected graph is a subset of edges no two of which share an endpoint. Use the property you derived in part (a) to obtain an  $O(mn^{1/2})$ -time algorithm for finding a matching of maximum cardinality in an undirected bipartite graph.

**Problem 3**

In class, we showed upper bounds on the number of push and relabel operations in the push-relabel algorithm, but did not specify the time complexity of a single operation. Give an implementation of the push-relabel algorithm described in class that has time complexity  $O(mn + k)$ , where  $k$  denotes the number of unsaturating pushes. Note that you should not use the fact that the upper bound derived on the number of unsaturating pushes was larger than that for saturating pushes and relabels; your algorithm should have a time complexity of  $O(mn + k)$  irrespective of the value of  $k$ . (**Hint:** For each vertex with non-zero excess flow, maintain an arbitrarily ordered list of edges leaving the vertex, and show that between any two consecutive relabels of the vertex, its edge list is sequentially scanned exactly once.)

**Problem 4**

Let  $M$  be an  $n_1 \times n_2$  matrix such that each entry  $M(i, j)$  is a non-negative real number and the sum of each row and each column is an integer. Show that there exists an  $n_1 \times n_2$  matrix  $N$  such that each entry  $N(i, j)$  is a non-negative integer and the sum of each row and each column is identical to that for  $M$ . Given  $M$ , how fast can you obtain  $N$ ?