6.854 Advanced Algorithms **Petar Maymounkov**
Problem Set 5 (October 11, 2005)
With: Benjamin Rossman, Oren Weimann, and Pouya Kheradpour

Problem 1a. Start with any graph and run $d+1$ blocking flows on it. Now the layered graph has at least d layers of vertices between source and sink. Hence, there must exist a pair of adjacent laywers that have no more than $2n/d$ vertices between the two of them. Think of this pair as a cut. The maximum number of edges going through this cut is $4n^2/d^2$, hence this is an upper bound on the max remaining flow.

This, we have a total of $O(d + n^2/d^2)$ blocking flows. Set $d = n^{2/3}$ to get a total running time of $O(mn^{2/3})$.

Problem 1b. Start with any graph and run $d+2p+1$ blocking flows on it. Now the layered graph has at least $d + 2p$ layers of vertices between source and sink. There are at least d pairs of adjacent layers that have no arbitray capacity vertex.

Among these pairs, there is at least one pair that has no more than $2n/d$ vertices between the to pairs. Consider this pair of edges as a cut. The maximum flow that could go through it is as most n^2/d^2.

Thus, the total maximum flow is $O(d+p+n^2/d^2)$. Set $d = n^{2/3}$, to get total runtim bound $O(m(p + n^{2/3}))$.

Another bound goes as follows. After $d + 2p$ blocking flows, the remaining augmenting paths all have length $d + 2p$, of which at most $2p$ vertices can be arbitrary capacity. Hence each augmenting path has at least d regular vertices. Thus there are at most m/d augmenting paths, each of capacity 1.

So, the total flow in the graph is $O(d+p+m/d)$, setting $d = \sqrt{m}$, gives us a total of $O(\sqrt{m}+p)$ blocking flows, and a total running time of $O(m^{3/2}+mp)$.

Problem 2b. Set up the following graph. Source s^*, vertices corresponding to the students $u_1, u_2, \ldots, u_s$, and vertices corresponding to the recitations $w_1, w_2, \ldots, w_r$, and sink t^*. Edges (s^*, u_i) of capacity 1 from the source to all students, edges (w_i, t^*) from the recitations to the sink of capacity k, finally, edges (u_i, w_j) from students to recitations of capacity 1, if student i can attend recitation j.

Problem 2a. When each recitation can accommodate only one student, this graph becomes a bipartire matching, and the fastest algorithm for it is $O(m\sqrt{n})$.

Problem 2c. In the general case when $k > 1$. Consider the residual graph after d blocking flows have been applied. All student vertices (similar to the bipartite case) have in or out degree 1. Therefore the path decomposition of the remaining flow is into paths that are disjoint on the student vertices. Since there are at most n student vertices, the maximum remaining flow is n/d. Hence total number of blocking flows is $O(d + n/d)$. Set $d = \sqrt{n}$, to obtain total running time $O(m\sqrt{n})$.

Problem 3. Begin by subtracting an appropriate integer from each matrix entry so that the entry falls in the interval $[0, 1)$. (This procedure can be

inverted at the end.) End up with a matrix where row and column sums are still integral, and matrix entries are in the range $[0, 1)$.

Let there be n rows and m columns. Represent the matrix by a bipartire graph with n nodes on the left l_i, m nodes on the right r_j, a source s and a sink t.

For each row $i \in [1, \ldots, n]$ in the matrix with sum g_i, add an edge (s, l_i) of capacity g_i. Also for every matrix entry a_{ij} add an edge (l_i, r_j) of capacity $\lceil a_{ij} \rceil$. (Note that $\lceil 0 \rceil = 0$.) Finally, for each column $j \in [1, \ldots, m]$ in the matrix with sum h_j, add an edge (r_j, t) of capacity h_j.

Under the given construction, the matrix represents a fractional maximum flow in the constructed bipartite graph. But we know that an integral flow exists (because all edges have integral values) and can be found using a MAX FLOW algorithm. The integral flow will have to correct all matrix entries by rounding them up or down, while not touching entries that are already integral.

Finally, we can transform the solution back to the original matrix by inverting the normalization procedure we did in the beginning.