

CS180 Homework 7

Zhehao Wang 404380075 (Dis 1B)

May 18, 2016

1 Min cut of the graph

(1, 2) (3, 4) (1, 3); this would cover the min cut (S, T), S, T non empty (one of the (s, t)s will satisfy s in S, t in T); as if not, at bottom level we'd know (1,2) (3,4) are each in the same S or T, and when we consider the first elements of two groups, we'd know all the nodes of the two groups are in the same group, until we reach top level where all nodes are in the same group which contradicts with S, T non empty. This is $O(m^2)$ per the description of the complexity of Ford Fulkerson $O(Ef)$, where E is m and f is the value of maxflow

2 Menger's theorem for vertices

Holds. We can consider only directed graph, as undirected graph can be seen as directed graph with edges both ways; for each node N, we split it into two nodes (s,t), all incoming edges to N now goes to s, all outgoing edges from N goes from t; each edge have capacity 1; do MaxFlow and we have the theorem. (After the split, because of each edge having capacity only 1, we guarantee node disjointness: each node is used at most once)

3 Deal cards and select

Bipartite graph, one part being the piles, the other part being the ranks; in/out degree of each makes Frobenius Hall Theorem hold, perfect matching exists.

4 Exam scheduling max flow

Bipartite graph, one part being the classes, the other part being cartesian product of (time, room); there exists an edge from classes to (time, room) only if the room is larger than the class. Do max flow by adding the s and t and setting weights accordingly, schedule exists if max flow = number of classes