Problem Set 4—Due Friday, October 26, 11:55PM

No late days for this assignment. All must be turned in by Friday

(25) Problem 1. Use network flows to find an efficient solution for the following problem:
   a) In a computer network there are \(n\) processors \(P_1, P_2, \ldots, P_n\), and \(m\) communication lines \(C_1, C_2, \ldots, C_m\). Each processor \(i\) has the ability to test \(t_i\) lines per day and there is a list \(L_i\) which contains the communication lines that processor \(i\) is able to test. Subject to these constraints we want to be able to test all the communication lines every day. A testing schedule determines for each processor the lines it should test. Find a testing schedule or determine that no schedule can test all lines in a single day.
   b) Same as a) but find the minimum number of days to test all lines (and a schedule which achieves it).
   c) Now we want to make sure that each line is tested twice each day, and it must be by two different processors.

(18) Problem 2 We consider a variation of the scaling algorithm for a network with integer capacities and where the maximum capacity arc out of the source is \(C^*\). In the standard algorithm we use \(\Delta = 2^k, 2^{k-1}, \ldots, 2, 1\) (where \(2^k < C^* \leq 2^{k+1}\)) and for each value of \(\Delta\) we find a max flow in \(G_f(\Delta)\) before moving on the next value. Now we instead use \(\Delta = 3^k, 3^{k-1}, \ldots, 3, 1\) where \(3^k < C^* \leq 3^{k+1}\).

This will change both the number of iterations of the outer loop (one for each \(\Delta\) value), and the maximum number of augmenting paths we need to find per value of \(\Delta\). Analyze each and discuss whether this is likely to lead to a better run time for the algorithm.

(24) Problem 3 We explore some issues related to the better method of finding augmenting paths discussed in class to improve the capacity scaling algorithm to \(O(mn)\) per scaling phase, and similar to a method used in 7.4 as part of the preflow-push algorithm.

   a) Prove the claim we made that if \((u, v) \in G_f\), then \(d(u) \leq d(v) + 1\) (at all times in the algorithm). Consider both the relabeling step, and the effects of edges being added or removed from \(G_f\).
   b) We can find a max-flow in a flow network where all capacities are one as follows: First run the shortest successive A-path algorithm (done in class on 10/18) until \(d(s) \geq n^{2/3}\). Let \(f_1\) be the flow at this point. Then switch to Ford-Fulkerson to convert the current flow \(f_1\) to a max-flow. Argue that this finds a maximum flow in time \(O(mn^{2/3})\) as long as the value of the flow \(f_1\) is within \(n^{2/3}\) of the max-flow value.
   c) Extra credit: Prove that \(f_1\) is within \(n^{2/3}\) of the max-flow value.

(22) Problem 4 Problem 39 page 440