Purdue University
CS 381

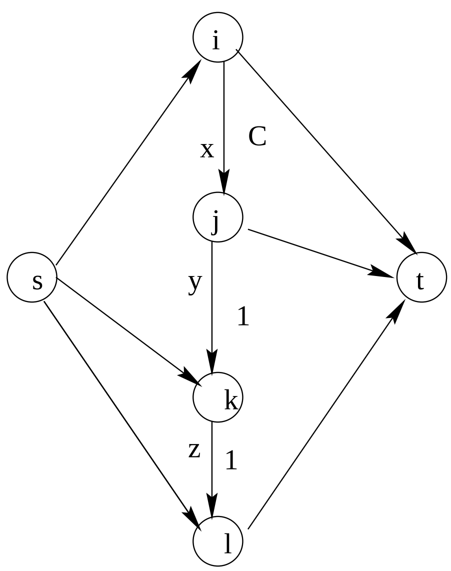
Gopal/Hambrusch
Spring 2008

Assignment 9

Due: Wednesday, April 23, 2008 (before class)

Final Exam: Friday, May 2, 3:20-5:20; check website for room yet to be assigned

1) (10 points) Consider the flow graph shown in the following figure.



The edges x, y, and z have capacities C, 1, and 1 respectively, where $C = \frac{\sqrt{5}-1}{2}$. All other edges have capacity 4.

1. Use the Ford-Fulkerson algorithm to compute the maximum flow that can be pushed between s and t in the above flow graph. Show the residual graph that results each time after an augmenting path is used.

2. Consider the following (potential) augmenting paths: $P1 : s \to i \to j \to k \to l \to t$, $P2 : s \to k \to j \to i \to t$, and $P3 : s \to l \to k \to j \to t$. Starting from zero flow on all edges, show the residual graph after applying the augmenting paths P1, P2, P1, P3 in that order.

   Show that after applying these four augmenting paths, the residual capacities of the edges x, y, and z are C^3, 1, and C^2 respectively. (Hint: The solution of the recurrence $t_{n+2} = t_n - t_{n+1}$, together with the initial conditions $t_0 = 1$ and $t_1 = C$, is $t_n = C^n$.) Does using the above augmenting paths repeatedly lead to termination?

2) (10 points) Consider the closest pair problem covered as an example of a divide-and-conquer algorithm in class and described in chapter 33.4 in the text. The closest pair of points in a set of

n points can be determined in $O(n \log n)$ time. Using one of the problems having an $\Omega(n \log n)$ lower bound stated in class and reduction, show that any algorithm solving the closest pair problem requires $\Omega(n \log n)$ time.

3) (20 points) Recall the decision version and the optimization version of the Hamiltonian Cycle (HC) problem.
(i) Show that the existence of a polynomial-time algorithm for the HC decision problem implies the existence of a polynomial time algorithm for the HC optimization problem.
(ii) Assume someone shows an $O(14n^{12} + 44n^5)$ time algorithm for the optimization version of the HC problem. What does this imply? Justify your answer.
(iii) The degree-restricted spanning-tree problem is (DRST) defined as follows: Given is an undirected graph G and an integer k. Determine whether G contains a spanning tree T such that every vertex of T has degree at most k. Show that DRST is NP-complete (by making a reduction from HC to DRST).

4) (10 points) The decision version of the 0/1 Knapsack problem is known to be NP-complete. Someone claims that the knapsack problem is easier to solve when one has more than one knapsack. Disprove this claim by showing that the Knapsack problem on two knapsacks is NP-complete.
The exact wording of the 2-knapsack decision problem is: Given are two knapsacks, each of capacity M, and m items, each having weight w_i and profit p_i, $1 \leq i \leq m$, and a profit P. Do there exist two disjoint sets of items S_1 and S_2 such that the items in S_1 fit into knapsack 1, the items in S_2 fit into knapsack 2, and the total profit is at least P?