College of Computer & Information Science
Northeastern University
CS5800: Algorithms

Summer 2011
Handout 1
31 July 2011

Makeup Final Exam

Problem 1. (10 points) Finding the last cell of an infinite array

The first n cells of an array A contain distinct integers. The remaining cells all contain some very large integer that we may think of as infinity (call it *maxint*). The array may be arbitrarily large (you may think of it as infinite), and you *do not know* n.

Give an algorithm to determine n in $O(\log n)$ time. For partial credit, you may give a less efficient algorithm.

Problem 2. (15 points) Directed path in a directed acyclic graph

Exercise 3.24, page 110 of text.

Problem 3. (15 points) Throwing a party

Exercise 5.27, page 166 of text.

Problem 4 (10 + 10 = 20 points) Short-answer questions on

For each of these questions, if your claim is "True", then give a brief proof; if your claim is "False", then provide a counterexample.

(a) For any connected weighted undirected graph G, any minimum spanning tree T of G, and vertices u and v of T, the unique path from u to v in T is the shortest path from u to v in G.

(b) Let T be a minimum spanning tree of an undirected weighted graph G. Let $C = (S, T)$ be a cut in G and let u be in S and v be in T. Then (u, v) is an edge of minimum weight in C.

Problem 5. (20 points) Document reconstruction

Exercise 6.4, page 192 of text.

Problem 6 (20 points) Job scheduling

A set of n clients has jobs that may be assigned to a set of m servers. Each client i has j_i jobs, each job being of the same size, and an associated set S_i of servers on which any of these jobs may be executed. Each server i can service at most s_i jobs. We would like to determine an assignment of jobs to servers such that the total number of jobs serviced is maximized. The assignment needs to only specify how many jobs of client i is assigned to server j, for all i and j. (Note that the j_i jobs of client i need not be assigned to the same server from S_i; they may be distributed among multiple servers.)

Design a polynomial-time algorithm based on network flows for the problem. Justify the correctness of your algorithm.