COS 226 Bob's Notes: Searching in Bitonic Arrays
Fall 2010
R. Tarjan, Preceptor

This, the first part of today's notes, discusses an extension of problem 6 on the midterm. The second part of today's notes will discuss the WordNet programming assignment.

Problem 6 asked for an algorithm to compute the maximum value in a bitonic array A of N numbers, where an array of numbers is bitonic if it consists of a strictly increasing sequence followed by a strictly decreasing sequence. This definition allows the possibility of the maximum value occurring twice, as the last in the increasing sequence and the first in the decreasing sequence. (In general, any value can occur twice, once in each sequence, but not more than twice.) The maximum can be found in ~lgN comparisons using binary search. Each step compares two adjacent numbers A[i] and A[i + 1]. If they are equal, they are both maximum. If A[i] is smaller, we restrict the search to indices at most i; if A[i +1] is bigger, we restrict the search to indices greater than i.

An extension of this problem is the following:

Precept Challenge Problem 1: Given a bitonic array A of N numbers and a number x, how many binary comparisons are needed in the worst case to determine if the array contains x?

The computation model for this problem is a comparison tree, in which each node is a comparison between two numbers in the array or between a number in the array and x, and each node has three children, corresponding to the three possible outcomes of the comparison: less than, equal, and greater than.

One can solve the problem in ~3lgN comparisons by finding the maximum in the array and then doing two binary searches, one on the increasing sequence and one on the decreasing sequence. The interesting question is whether one can do better. I conjecture that the problem can be solved in ~2lgN comparisons, and that this number is required in the worst case. Your challenge is to prove or disprove one or both halves of this conjecture. We will give class credit to anyone who comes up with a full or partial solution. Working on this problem is entirely optional and is intended as an enrichment exercise for any of you who are looking for a little extra challenge. Please turn in your solution, if any, in precept on November 19.