CS140: Algorithms

Spring 2001

Assignment 5

This assignment is due by the start of class on Thursday, Feb. 8. You may collaborate with your classmates to complete this assignment but you must reference your collaborators. In addition, you must do your own write-up. This assignment is worth 100 points (65 points if handwritten).

1. $S = S[0], S[1], \ldots, S[n-1]$ is an array of distinct integers. S is said to be *unimodal* if there is some index i (where $0 \leq i \leq n-1$) such that $S[0], ..., S[i]$ is a strictly increasing sequence and $S[i], ..., S[n-1]$ is a strictly decreasing sequence. For example, $S = 2, 4, 8, 7, 6, 5, 3, 1, 0, -1$ is unimodal but $S = 2, 3, 1, 4$ is not.

   Consider the problem of finding the maximum element in a unimodal array S when $n = 2^m$. Assume an algorithm is only allowed access to S through queries of the form "Is $S[j] < S[k]$?" Give the best upper and lower bounds you can on the number of queries needed to solve this problem. Your answer should not use big-O notation; i.e. you can get exact bounds for this problem. Prove your answers; i.e. give an algorithm that achieves the upper bound and an adversary that achieves the lower bound.

2. In class we showed that any comparison-based sorting algorithm is $\Omega(n \lg n)$. This bound does not necessarily hold if we restrict the input in some way. For example, the input may be "semi-sorted" to begin with. Consider the following definitions of "semi-sorted" and determine whether the problem of sorting an input that satisfies the definition can break the $\Omega(n \lg n)$ bound. Prove your answer.

   (a) The input array is a min-heap.

   (b) Each element in the input array is within 5 positions of its sorted position. To be more precise, the rank of element s_i in the input list is between $i - 5$ and $i + 5$. (Assume the input integers are distinct.)