
CS 473: Undergraduate Algorithms, Spring 2009

HBS 9

1. Prove that any algorithm to merge two sorted arrays, each of size n , requires at least $2n - 1$ comparisons.

2. Suppose you want to determine the largest number in an n -element set $X = \{x_1, x_2, \dots, x_n\}$, where each element x_i is an integer between 1 and $2^m - 1$. Describe an algorithm that solves this problem in $O(n + m)$ steps, where at each step, your algorithm compares one of the elements x_i with a *constant*. In particular, your algorithm must never actually compare two elements of X ! *[Hint: Construct and maintain a nested set of 'pinning intervals' for the numbers that you have not yet removed from consideration, where each interval but the largest is either the upper half or lower half of the next larger block.]*

3. Let P be a set of n points in the plane. The staircase of P is the set of all points in the plane that have at least one point in P both above and to the right. Prove that computing the staircase requires at least $\Omega(n \log n)$ comparisons in two ways,
 - (a) Reduction from sorting.
 - (b) Directly.