

Homework 6, Due Friday, November 13, 2015

Problem 1 (10 points):

Suppose you are choosing between the following three algorithms:

- Algorithm A solves the problem by dividing it into five subproblems of half the size, recursively solves each subproblem, and then combines the solution in linear time.
- Algorithm B solves problems of size n by recursively solving two subproblems of size $n - 1$, and then combines the solution in constant time.
- Algorithm C solves the problem by dividing it into nine subproblems of one third the size, recursively solves each subproblem, and then combines the solutions in quadratic time.

What are the running times of each of these algorithms?

Problem 2 (10 points):

Given an array of elements $A[1, \dots, n]$, give an $O(n \log n)$ time algorithm to find a majority element, namely an element that is stored in more than $n/2$ locations, if one exists. Note that the elements of the array are not necessarily integers, so you can only check whether two elements are equal or not, and not whether one is larger than the other. HINT: Observe that if there is a majority element in the whole array, then it must also be a majority element in either the first half of the array or the second half of the array. (This is also exercise 3, page 246 from the text, without the annoying storyline.)

Problem 3 (10 points):

Page 248, Exercise 5.

Problem 4 (10 points):

Page 248, Exercise 6.