

Homework 6, Due Wednesday, November 9, 2016

Problem 1 (10 points):

Solve the following recurrences:

a)

$$T(n) = \begin{cases} T(\frac{n}{2}) * T(\frac{n}{2}) \\ 2 \end{cases} \quad \text{if } n \leq 1$$

b)

$$T(n) = \begin{cases} T(n-1) * T(n-1) \\ 2 \end{cases} \quad \text{if } n \leq 1$$

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 6.

Problem 4 (10 points):

Page 248, Exercise 7.