

COSC 311: ALGORITHMS  
HW2: DIVIDE-AND-CONQUER

Due Friday, October 12 in class

## 1 Submission and Collaboration

This assignment is due on Friday, October 12 at the start of class; bring a hard copy of your typed responses to submit in class. **Each problem must be on a separate sheet of paper, with your name and section number at the top of the page.** If you use more than one sheet of paper for an individual problem, please staple together those pages. There is no need for you to rewrite the problems on your submission.

In accordance with this course's intellectual responsibility policy, you are welcome to discuss the problems with other students who are currently taking the course, but you must write up your solutions individually. Please note at the top of your submission with whom you discussed each problem.

## 2 Problems

### Problem 1: Using the Master Theorem.

For each of the following recurrences, check whether the Master Theorem applies. If it does, give the asymptotic class of  $T(n)$ . If not, explain why not.

(a)  $T(n) = 9T(n/3) + 4n^2$

(b)  $T(n) = 8T(n/2) + \frac{2n^3}{\lg n}$

(c)  $T(n) = 2T(n/2) + n\sqrt{n}$

(d)  $T(n) = 3T(n/3) + f(n)$  where

$$f(n) = \begin{cases} n^2 & \text{if } n \text{ is a power of } 9 \\ 6n^2 & \text{otherwise} \end{cases}$$

## 2) Probing binary trees.

Consider an  $n$ -node complete binary tree (by “complete,” we mean that every level is full—so  $n = 2^d - 1$  for some positive integer  $d$ ). Each node of the tree  $x$  stores some value  $v_x$ . You can assume that all of the values are distinct, i.e.,  $v_x \neq v_y$  for all  $x \neq y$ . A node  $x$  is considered a *local minimum* if its value  $v_x$  is less than the value  $v_y$  for all nodes  $y$  that are connected to  $x$  by an edge (i.e.,  $v_x$  is smaller than the values of  $x$ ’s parent and children if they exist).

You can look up the value of a node  $x$  by *probing* the node. Assume that probing is a constant-time operation.

(a) Describe an algorithm that finds a local minimum in time  $O(\lg n)$ . You may write pseudocode or clearly state the steps of your algorithm; your goal is for your description to be precise and unambiguous.

(b) Explain why your algorithm is correct. You may want to use induction or some other form of reasoning. Your goal here is to convince a classmate that your algorithm does the right thing.

(c) Write a recurrence for the runtime of your algorithm,  $T(n)$ , and solve your recurrence using the Master Theorem.

## 3) Database queries.

You are interested in analyzing some hard-to-obtain data from two separate databases. Each database contains  $n$  numerical values—so there are  $2n$  values total—and you may assume that no two values are the same. You want to determine the median of this set of  $2n$  values, which we will define to be the  $n$ th smallest value.

The only way you can access these values is through queries to the database. In a single query, you can specify a value  $k$  to one of the two databases, and the chosen database will return the  $k$ th smallest value that it contains. Since queries are expensive, you would like to compute the median using as few queries as possible.

(a) Describe an algorithm that finds the median value using at most  $O(\log n)$  queries. You may write pseudocode or clearly state the steps of your algorithm; your goal is for your description to be precise and unambiguous.

(b) Explain why your algorithm is correct. You may want to use induction or some other form of reasoning. Your goal here is to convince a classmate that your algorithm does the right thing.

(c) Write a recurrence for the runtime of your algorithm,  $T(n)$ , and solve your recurrence using the Master Theorem.