CS 312: Algorithms **Fall 2018**

Fourth Hour 8

Your Name: _____ Collaborators: _____

You will be randomly assigned groups to work on these problems in discussion section.

Problem 1. Recursion Tree. Consider the following recurrence with $q = 3$:

$$T(n) \leq qT(n/2) + cn$$
$$T(2) \leq c$$

This recurrence describes an algorithm with a particular structure. On an input of size n,

- How many recursive calls does the algorithm make?
- What is the problem size for the recursive calls?
- How much work is done outside of the recurrence?

Now, draw a recursion tree for this recurrence. Like we did in class, calculate the following quantities for each level i in the recursion tree:

- Problem size at level i
- Work per problem at level i (outside of recursive calls)
- Number of problems at level i
- Total work done at level i

Also compute the depth d of the recursion tree.

Now, add up the work per level for all levels i of the recursion tree to get an expression for the total work, which is the running time of an algorithm with this recurrence. You can leave your expression as a summation. When you finish this, move on to the next problem. When all groups have made enough progress, we will work together to simplify the summation.

Problem 2. Database Medians. *This problem will be on next week's homework.*

You are working as a programmer for Mount Holyoke administration, and they ask you to determine the median GPA for all students. However, their system is really out of date, and student GPAs are stored in two different databases, one for students with last names that begin with A–L, and another for students with last names that begin with M–Z. Assume there are n students in each database, so there are $2n$ students total. You'd like to determine the median of this set of $2n$ values, which we will define here to be the nth smallest value.

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

Give an algorithm that finds the median value using at most $O(\log n)$ queries.