CSCI-570 Homework 3

Chunho Lin* (3226964170)
chunholi@usc.edu

Due Date: Tuesday, October 7th, 11:59pm

1. **Problem 1.**

   The recurrence $T(n) = 12 \cdot T(\frac{n}{3}) + n^2$ describes the running time of an algorithm ALG. A competing algorithm ALG$'$ has a running time of $T'(n) = a \cdot T'(n/9) + n^2$. What is the range of the values of a such that ALG$'$ is asymptotically faster than ALG?

   [6 points]

   Proof. ■

2. **Problem 2. [Think through]**

   What is the runtime of an algorithm that solves problems of size n by recursively solving two subproblems of size $n-1$ and then combines the solutions in constant time?

   [3 points]

   Think through. ■

3. **Problem 3.**

   Consider a divide and conquer algorithm which solves problems by dividing them into five subproblems, each of size $\frac{n}{2}$, recursively solving each subproblem, and then combining the solutions in linear time. Write down the runtime recurrence relation of this algorithm, then calculate its runtime by unrolling its recursion tree and using the runtime recurrence. (Providing the runtime based on applying the Master Theorem will not be accepted as an answer.)

   [4 points]

   Proof. ■

4. **Problem 4.**

   Given an array A of n positive integers (with repetition allowed), determine if the array has a majority element, that is, a number that occurs in the array more than $\frac{n}{2}$ times. Provide a $O(n \cdot logn)$ runtime divide and conquer algorithm that returns the value of the majority element if there is one, and -1 if there isn't.

   Examples:
   Input: $A = [1, 2, 9, 2, 2, 7, 6, 2, 2]$, Output: 2;
   Input: $A = [1, 2, 3, 6, 5, 6, 1, 8, 1]$, Output: -1.

[10 points]

Proof. ■

5. **Problem 5.**

   Consider a two-dimensional array A of size $n \times n$ filled with integers. In the array each row is sorted in ascending order and each column is also sorted in ascending order. Our goal is to determine if a given value k exists in the array. Design a divide-and-conquer algorithm that solves this problem based on checking "the middle element" of the array (i.e. "$A[n/2][n/2]$") and dividing the problem based on its value. What is the runtime of your algorithm? (It should be faster than $O(n^2)$.)

   [8 points]

   Proof. ■