CSE 431/531: Algorithm Analysis and Design Fall 2021

Homework 3

Instructor: Shi Li **Deadline: 11/4/2021**

Your Name: _____ Your Student ID: _____

Problems	1	2	3	Total
Max. Score	20	30	30	80
Your Score				

Problem 1. For each of the following recurrences, use the master theorem to give the tight asymptotic upper bound.

(a) $T(n) = 4T(n/3) + O(n)$. $T(n) = O(\_\_\_)$.

(b) $T(n) = 3T(n/3) + O(n)$. $T(n) = O(\_\_\_)$.

(c) $T(n) = 4T(n/2) + O(n^2\sqrt{n})$. $T(n) = O(\_\_\_)$.

(d) $T(n) = 8T(n/2) + O(n^3)$. $T(n) = O(\_\_\_)$.

Problem 2. We consider the following problem of counting stronger inversions. Given an array A of n positive integers, a pair $i, j \in \{1, 2, 3, \cdots, n\}$ of indices is called a *strong inversion* if $i < j$ and $A[i] > 2A[j]$. The goal of the problem is to count the number of strong inversions for a given array A.

Give a divide-and-conquer algorithm that runs in $O(n \log n)$ time to solve the problem. Write down the recurrence for the running time, and use the master theorem to show that the running time is indeed $O(n \log n)$.

Problem 3. Given two sorted arrays A and B with total size n, and a positive integer $k \leq n$, you need to design an $O(\log n)$-time algorithm that outputs the k-th smallest number in the union of A and B. You need to prove that the running time of your algorithm is indeed $O(\log n)$.

For example, if $A = [3, 5, 12, 18, 50], B = [2, 7, 11, 30]$, and $k = 4$ then you need to output 7 since the union of A and B is $[2, 3, 5, 7, 11, 12, 18, 30, 50]$ after sorting.