Quiz #1
Introduction to Algorithms
601.433/633
Spring 2020

You must submit your solutions by Wednesday, April 8th, 10am. Late submissions will NOT be accepted. You may submit handwritten answers – you will have to scan/photograph them, convert it to a pdf and upload it to Gradescope.

1 Problem 1 (50 points)

For each statement below explain if it is true or false and in a couple of sentences provide an explanation for your answer. Be as mathematically precise as you can in your explanation. The base of $\log$ is 2 unless stated otherwise.

1. $2^{n^2} = \Theta(3^{n+\sqrt{n}})$
2. $n! = \omega(2^n)$
3. Let f be positive function. Then $f(n) = O((f(n))^2)$.

2 Problem 2 (50 points)

Resolve the following recurrences in terms of a big-Θ bound. You may assume that $T(0) = T(1) = 1$. Provide a proof for the bound you give. If appropriate, you may invoke the Master Theorem (and the appropriate case).

1. $T(n) = T(n-1) + 2T(n-2)$
2. $T(n) = 7T(n/15) + n^5$

3 Problem 3 (50 points)

A sequence $a_1, a_2, \ldots, a_n$ has a special element if more than half of the elements in the sequence are the same. For example, 3 is a special element in the sequence $7, 3, 3, 3, 1, 3, 3, 4, 5, 3$. On the other hand, the sequence $5, 4, 1, 1, 2, 3, 2, 3, 6$ has no special element. Give a divide and conquer algorithm that runs in time $O(n \log n)$ and returns a special element in a sequence of n numbers or returns $None$ if no such element exists. Prove the correctness of your algorithm and prove that its runtime is $O(n \log n)$. (Note: there exists an $O(n)$ algorithm to solve this problem that doesnt make use of divide and conquer if you figure it out, you may prove its correctness and runtime instead.)

4 Problem 4 (50 points)

You are given n tasks for a machine. Task i is described by $l_i = [a_i, b_i]$ on the real line, where a_i, b_i are real numbers, $a_i \leq b_i$ and $1 \leq i \leq n$. Give an algorithm that computes the total times of this set of tasks, that is, the length of $\cup_{i=1}^{n} l_i$ in $O(n \log n)$ time.

For example, for the set of tasks $\{[1, 3], [2, 4.5], [6, 9], [7, 8]\}$, the total time is $(4.5 - 1) + (9 - 6) = 6.5$.

Make sure to prove the correctness and running time of your algorithm.

5 Problem 5 (50 points)

Suppose that you have a set of n integers, $A = \{a_1, ..., a_n\}$, each of them is between 0 and K (inclusive). Your goal is to find a partition of A into two sets S_1 and S_2 (so $S_1 \cup S_2 = A$ and $S_1 \cap S_2 = \emptyset$) that minimizes $|W(S_1) - W(S_2)|$ where $W(S)$ denote the sum of integers in S. Your algorithm's running time should be polynomial in n and K.

Make sure to prove the correctness (mainly the optimal substructure property) and running time.