(CS1.301) Algorithm Analysis & Design (Monsoon 2024)

Practice Problem Set 3

Instructions:

- The following problem set is not graded and is for practice.
- Some of the problems will be covered in the tutorial.

Question 1

Given a set, P , of n teams in some sport, a round-robin tournament is a collection of games in which each team plays each other team exactly once. Such round-robin tournaments are often used as the first round for establishing the order of teams (and their seedings) for later single- or double- elimination tournaments. Design an efficient algorithm for constructing a round-robin tournament for a set, P , of n teams assuming n is a power of 2.

Question 2

Imagine you're searching for the best smartphone based on two criteria: battery life and camera quality. Each smartphone is represented by a point in a two-dimensional space, where one axis represents battery life and the other represents camera quality. You want to find all smartphones that are optimal in terms of these criteria — meaning there are no other smartphones in the set that have both a longer battery life and a better camera quality. These optimal smartphones form what we call the "maxima set."

To efficiently identify this maxima set among a given set of smartphones, you would employ a divide-and-conquer strategy that helps you compare and eliminate smartphones that are dominated by others based on battery life and camera quality.

Question 3

Consider the following algorithm given an array as input. Divide the input array into $\lfloor \frac{n}{5} \rfloor$ blocks, each containing exactly 5 elements, except possibly the last. (If the last block isn't full, just pad it with a few 1s.) Then compute the median of each block by brute force, collect those medians into

a new array $M[1, \ldots, \lfloor \frac{n}{5} \rfloor]$, and then recursively compute the median of this new array. Show that this algorithm computes an element that is at least the $\frac{3}{10}^{\text{th}}$ largest and at most the $\frac{7}{10}^{\text{th}}$ largest in the original array.

Question 4

Given a sorted array of distinct integers $A[1 \ldots n]$, we want to determine whether there exists an index i such that $A[i] = i$. Design a divide-and-conquer algorithm that runs in time $O(\log(n))$.

Question 5

You are given an array of n elements, and you notice that some of the elements are duplicates; that is, they appear more than once in the array. Show how to remove all duplicates from the array in time $O(n \log n)$.

Question 6

A k-way merge operation. Suppose you have k sorted arrays, each with n elements, and you want to combine them into a single sorted array of kn elements.

(a) Here's one strategy: Using the merge procedure from Section 2.3, merge the first two arrays, then merge in the third, then merge in the fourth, and so on. What is the time complexity of this algorithm, in terms of k and n?

(b) Give a more efficient solution to this problem, using divide-and-conquer.