Design and Analysis of Algorithms

Spring 2024

Assignment 1

Submission Date: Feb 26, 2024

Question 1

Suppose each member of the national assembly of banana republic wants to pass his/her resolution. The constitution of banana republic pass only those resolutions that are supported by more than half members of the assembly. None of the members of the assembly wants to disclose its resolution until it is passed by the assembly. Now assume that they have a device that can tell whether two members have same resolution or not. Your task is to device an algorithm that uses this device no more than O(nlgn) time and find out whether there is some resolution that this assembly can pass or not. Here n is the number of members in the national assembly.

Question 2

Suppose a coin is tossed n number of times and the result of tosses is stored in an array such that all occurrences of heads are followed by all occurrences of tail. Your task is to device an algorithm that finds the number of occurrences of tail in O(lgn) time.

Question 3

Suppose you are given an array of 3 colors: red, orange and blue. Each color may occur multiple times in this array. You task is to device an in place O(n) algorithm that sort this array such that first all reds are placed followed by orange and then green. Here n is the size of the array. Please note that you are not allowed to count the number of occurrences of these colors. All you can do is just compare the array elements with red, orange or green.

Question 4

Suppose you are given an array A with n entries, with each entry holding a distinct number. You are told that the sequence of values A[1], A[2], ..., A[n] is unimodal. For some index p between 1 and n, the values in the array entries increase up to position p in A and then decrease the remainder of the way until position n. You'd like to find the "peak entry" p without having to read the entire array in fact, by reading as few entries of A as possible. Show how to find the entry p by reading at most O(log n) entries of A.

Question 5

Consider an n − node complete binary tree T , where $n = 2^d - 1$ for some d. Each node v of T is labeled with a real number x_v. You may assume that the real numbers labeling the nodes are all distinct. A node v of T is a local minimum if the label x_v is less than the label x_w for all nodes w that are joined to v by an edge. You are given such a complete binary tree T, but the labeling is only specified in the following implicit way: for each node v, you can determine the value x_v by probing the node v. Show how to find a local minimum of T using only O(log n) probes to the nodes of T.

Question 6

Solve the following recurrences and compute the asymptotic upper bounds. Assume that T(n) is a constant for sufficiently small n. Make your bounds as tight as possible.

a. $T(n) = 7T\left(\frac{n}{2}\right) + n^2$
b. $T(n) = T(n-2) + \sqrt{n}$
c. $T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + O(n)$
d. $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$

Note: Please submit the C++ code for the algorithms you design in Q1-Q5