081203M04001H - Algorithm Design and Analysis

Assignment 1

October 16, 2020

Notice:

1. The assignment contains two parts.
   (a) For problems 1-6, please submit your answer in hard copy AND submit a digital version to UCAS website `http://sep.ucas.ac.cn`.
   Hard copy should be submitted before 9 am. October 30 and digital version should be submitted before 11 pm. October 30.
   (b) For problems 7-8, you need finish them on the website `http://theory.ict.ac.cn/grad_oj` before 10 am. October 23.
2. You can choose **three** from problems 1-6.
3. For problems 1-6, you should do at least the following things:
   (a) Describe your algorithm in natural language **AND** pseudo-code;
   (b) Draw a "subproblem reduction graph", where nodes represent subproblems, and edges describe the "reduction relationship" between them for every problem you choose in problems 1-6;
   (c) Prove the correctness of your algorithm;
   (d) Analyse the complexity of your algorithm.
4. For problems 7-8, you can implement your algorithm in C/C++/Java/Python/Pascal.

1 Divide and Conquer

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand. (i.e., $[0, 1, 2, 4, 5, 6, 7]$ is an ascending array, then it might be rotated and become $[4, 5, 6, 7, 0, 1, 2]$.) How to find the minimum of a rotated sorted array?
(*Hint:* All elements in the array are distinct.)

For example, the minimum of the rotated sorted array $[4, 5, 6, 7, 0, 1, 2]$ is 0.

Please give an algorithm with $O(\log n)$ complexity, prove the correctness and analyze the complexity.

2 Divide and Conquer

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 $O(logn)$ probes to the nodes of T.

3 Divide and Conquer

Given an integer array, one or more consecutive integers in the array form a sub-array. Find the maximum value of the sum of all subarrays.

Please give an algorithm with O(nlogn) complexity

4 Divide and Conquer

Given an array of integers nums sorted in ascending order, find the starting and ending position of a given target value. If the target is not found in the array, return [-1, -1]. For example, if the array is $[5, 7, 7, 8, 8, 10]$ and the target is 8, then the output should be $[3, 4]$.

Your algorithm's runtime complexity must be in the order of $O(\log n)$, prove the correctness and analyze the complexity.

5 Divide and Conquer

Given a convex polygon with n vertices, we can divide it into several separated pieces, such that every piece is a triangle. When $n = 4$, there are two different ways to divide the polygon; When $n = 5$, there are five different ways.

Give an algorithm that decides how many ways we can divide a convex polygon with n vertices into triangles.

6 Divide and Conquer

Recall the problem of finding the number of inversions. As in the course, we are given a sequence of n numbers $a_1, ..., a_n$, which we assume are all distinct, and we define an inversion to be a pair $i < j$ such that $a_i > a_j$.

We motivated the problem of counting inversions as a good measure of how different two orderings are. However, one might feel that this measure is too sensitive. Let's call a pair a significant inversion if $i < j$ and $a_i > 3a_j$. Given an $O(n \log n)$ algorithm to count the number of significant inversions between two orderings.

7 Divide and Conquer

Given two integers m and $n(n > 0)$, find the integer represented by the last three digits of m^n.

For example, if $m = 3$ and $n = 9$, the returned number should be $683(3^9 = 19683)$.

Please give an algorithm with $O(\log n)$ complexity.

Note: In a computer program, when m or n is large, exponential calculations may cause data overflow. Your algorithm should be able to avoid this problem.

INPUT:

Line 1:two integers m n, split by space

OUTPUT:

one integer

8 Divide and Conquer

Given N points on the plane. Find the Kth closest point to the origin $(0, 0)$. (Here, the distance between two points on a plane is the Euclidean distance.)

INPUT:

Line 1:N K

Others:the coordinates of point

OUTPUT:

the Kth point