091M4041H - Algorithm Design and Analysis

Assignment 1

September 27, 2019

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 11 and digital version should be submitted before 11 pm. October 11.
   (b) For problems 7-8, you need finish them on the website `http://theory.ict.ac.cn/grad_oj` before 11 pm. October 18.
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.)

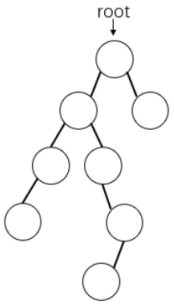
For example, the minimum of the rotated sorted array $[4, 5, 6, 7, 0, 1, 2]$ is 0.

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

2 Divide and Conquer

Given a binary tree, suppose that the distance between two adjacent nodes is 1, please give a solution to find the maximum distance of any two node in the binary tree.

Note: For BinaryTree, each node has three properties, a int value and two TreeNode pointers to children. You can assume that the INPUT is the root TreeNode of the tree. For example, the maximum distance of any two nodes in the below binary tree is 5.



3 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 $O(\log n)$ *probes* to the nodes of T.

4 Divide and Conquer

Suppose now that you're given an $n \times n$ grid graph G. (An $n \times n$ grid graph is just the adjacency graph of an $n \times n$ chessboard. To be completely precise, it is a graph whose node set is the set of all ordered pairs of natural numbers (i, j), where $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant n$; the nodes (i, j) and (k, l) are joined by an edge if and only if $|i - k| + |j - l| = 1$.)

We use some of the terminology of problem 3. Again, each node v is labeled by a real number x_v; you may assume that all these labels are distinct. Show how to find a local minimum of G using only $O(n)$ probes to the nodes of G. (Note that G has n^2 nodes.)

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 $a_1, \cdots, a_n$, 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 an array of size n, find the majority element. The majority element is the element that appears more than $\lfloor \frac{n}{2} \rfloor$ times.You may assume that the array is non-empty and the majority element always exist in the array.

INPUT:

Line 1: the length of array.

Line 2: the all elements in array and split by spaces

OUTPUT:

Line 1: A single integer that is the majority element.

8 Divide and Conquer

Given a 2d $m * n$ grid map of '1's (land) and '0's (water), count the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

INPUT:

Line 1: m and n.

Line 2: '1' or '0' in grid and split by spaces

OUTPUT:

Line 1: number of islands.