CS 577: Introduction to Algorithms — Divide and Conquer Review

Problems

1. For this problem, a subtree of a binary tree means any connected subgraph. A binary tree is complete if every internal node has two children, and every leaf has exactly the same depth. Describe and analyze a recursive algorithm to compute the $largest\ complete\ subtree$ of a given binary tree. Your algorithm should return the root and depth of the subtree.

2. Given an array $A[1 \ldots n]$ of integers, develop an algorithm to count the number of significant inversions where given i and j as indexes in the array, it satisfies $i < j$ and $A[i] > 2 * A[j]$. A sorted array would have an inversion count of 0.

3. Consider a n by n matrix of integers where each row is sorted in increasing order from top to bottom, and each column is sorted in increasing order from left to right. Design and analysis a divide and conquer algorithm similar to binary search to determine whether a given integer lies in the matrix, and if so return its row and column number assuming both row and column start with index 0 and end at index n-1. Assume that all integers in matrix are distinct

4. In this problem, you are given an array $A[0 \cdots n-1]$ of n distinct numbers sorted in increasing order and then right-shifted circularly by some unknown number of places. Your goal is to figure out the amount of shift, equivalently, the position of the smallest element in the array. (Note: we are indexing the array from 0 to $n-1$ here.) For example, if you are given the array $[5, 7, 9, 10, 1, 3, 4]$, your answer should be 4, which is the index of the smallest element 1 and is also the number of places the sorted array $[1, 3, 4, 5, 7, 9, 10]$ would need to be right-shifted to produce the given array. Develop a divide and conquer algorithm for this problem. Hint: Try writing a procedure to find the minimum element of the list.