Tutorial 2, Design and Analysis of Algorithms, 2023

1. Given two sorted arrays A and B, find the k-th smallest element in the final sorted array. Design a divide and conquer algorithm to solve the problem. Justify the time complexity.

2. You are given a unimodal array of n distinct elements, meaning that its entries are in increasing order up until its maximum element, after which its elements are in decreasing order. Give a divide and conquer algorithm to compute the maximum element of a unimodal array that runs in $O(logn)$ time.

3. Given a set $P = \{p_1, p_2, \ldots, p_n\}$ of n points in $\mathbb{R}^2$, where $p_i = (x_i, y_i)$. Let s_{ij} denotes the slope of the line segment joining p_i and p_j and $S = \{s_{ij} \mid 1 \leq i < j \leq n\}$. Basically, S contains all the slopes. Thus, $|S| = O(n^2)$. The objective is to compute a count on the number of elements of S that lie in the interval $[a, b]$, here a and b are real numbers. Devise an $O(n \log n)$ time algorithm for it, rather than the obvious $O(n^2)$ time. *(Hint: 1. Use geometrical property-dual of line and point, 2. application of the inversion problem.)*

4. Recall the problem of finding the number of inversions. As discussed in the class, we are given a sequence of n numbers $a_1, \ldots, a_n$, which we assume are all distinct, and we define an inversion to be a pair $i < j$ such that $a_i > a_j$. However, one might feel that this measure is too sensitive. Let's call a pair a significant inversion if $i < j$ and $a_i > 2a_j$. Give an $O(nlogn)$ algorithm to count the number of significant inversions with new measure.

5. Find 7499×9274 using *Karatsuba's Divide and Conquer Integer Multiplication Algorithm.*

6. Devise a divide and conquer algorithm for multiplying n-digit long integer to a single digit. Justify the time complexity. *(eg. 4 digit: 1234 multiplied by 5).*

7. Devise a divide and conquer algorithm for adding two n-digit long integers. Justify the time complexity.

8. **Maximum Contiguos Subsequence Sum**

   Input: A sequence of n integers.

   Ouput: $max(\sum_{i=b}^{i=e} a_i \mid 1 \leq b \leq e \leq n)$.

   Eg $A = -3, 1, 3, -3, 4, -7$, the maximum contiguous subsequence is $1, 3, -3, 4$, the result is 5.

   Devise an efficient Divide and Conquer algorithm for the problem.

9. Suppose we modify the Select algorithm (as discussed in class) by breaking the elements into groups of 7, rather than groups of 5. (Use the median of medians as the pivot element, as before.) Does this modified algorithm also run in $O(n)$ time? What if we use groups of 3? What if we use groups of 9?