Problem Sheet [1]

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 $O(\log n)$ time.

3. In an infinite array, the first n cells contain integers in sorted order and the rest of the cells are filled with ∞. Present an algorithm that takes x as input and finds the position of x in the array in $O(\log n)$ time. You are not given the value of n.

4. 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 maximum slope in S in $O(n \log n)$ time, rather than the obvious $O(n^2)$ time. Devise an algorithm for the above mentioned objective. *(Hint: 1. Use geometrical property, 2. the desired time complexity is another clue.)*

5. 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 $O(n \log n)$ algorithm to count the number of significant inversions with new measure.

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

7. Devise a divide and conquer algorithm for multiplying n-digit long integer to a single digit. Do it in $O(n \log n)$. Justify the time complexity. *(eg. 4 digit: 1234 multiplied by 5).*

8. Devise a divide and conquer algorithm for adding two n-digit long integers. Do it in $O(n \log n)$. Justify the time complexity.

9. **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.

10. Given a set of n intervals $I = [a_1, b_1], [a_2, b_2], \ldots [a_n, b_n]$. Here $a_i < b_i$ for all $i = 1$ to n. Devise a Divide and Conquer algorithm to compute the length of the biggest overlap between any two intervals in $O(n \log n)$ time. Justify the time complexity. For eg, $[1, 7]$ overlaps with $[3, 9]$, and the length of the overlap between them is $7 - 3 + 1 = 4$.

[1] Prepared by Pawan K. Mishra