Practice Problem Set – Basic Graphs and Divide and Conquer

1. Given an undirected graph $G = (V, E)$ and two specified vertices u and v. Design an algorithm that computes the number of shortest paths between u and v in G. Your algorithm does not have to list all shortest paths between u and v in G, just the number of shortest paths between u and v suffices. By definition, a shortest path between u and v is a path between u and v with minimum number of edges.

2. Suppose you are given two sets of distinct points, one set $\{p_1, \ldots, p_n\}$ on line $y = 0$ and the other set $\{q_1, q_2, \ldots, q_n\}$ on line $y = 1$. Create a set of n line segments by connecting each point p_i with q_i. Describe a divide and conquer algorithm to determine the number of these line segments that intersect in $O(n \log n)$-time.

3. Given an array A of n distinct elements sorted in increasing order. Design an algorithm that determines if there is an index i such that $A[i] = i$.

4. One ordered pair (u_1, v_1) dominates (u_2, v_2) if $u_1 \geq u_2$ and $u_2 \geq v_2$. Given a collection S of n ordered pairs, an ordered pair (u^*, v^*) is called *Pareto optimal* of S if there is no order pair $(u', v') \in S$ that dominates (u^*, v^*).

   Design an efficient algorithm that takes a collection of n ordered pairs, and outputs the collection of all Pareto optimal pairs in this list.

   Explain the running time and justify with formal proof why the algorithm works correctly.

5. An array $A[1, 2, \ldots, n]$ of n distinct numbers is *bitonic* if there are unique indices i and j such that $A[(i-1) \mod n] < A[i] > A[(i+1) \mod n]$ and $A[(j-1) \mod n] > A[j] < A[(j+1) \mod n]$. In other words, a bitonic sequence either consists of an increasing sequence followed by a decreasing sequence or can be circularly shifted to become so.

   For example 4, 6, 9, 8, 7, 5, 1, 2, 3 is bitonic but 3, 6, 9, 8, 7, 5, 1, 2, 4 is not bitonic.

   Let A be a bitonic array of n distinct numbers. Design and analyze an algorithm to find the smallest element of A in $O(\log_2 n)$-time.

6. Suppose that there are two complex numbers $a + ib$ and $c + id$. Design an algorithm that multiplies the two above mentioned complex numbers using at most three multiplications.

In short, your algorithm should take a, b, c, d as input and compute $ac - bd$ as well as $ad + bc$ using at most three multiplications.

7. Given an array A of n numbers, design an algorithm that identifies if there is any number that appears twice in A. Your algorithm must run in time asymptotically faster than $O(n^2)$-time. Suppose that the numbers in array A are integers in the set $\{1, 2, \ldots, 4n\}$. Can you design an $O(n)$-time algorithm for that?

8. The frequency of a number in an array is the number of times it appears in the array. Given an array A of n numbers, design an algorithm that finds a number appearing the maximum number of times. Explain the time complexity of your algorithm.

9. Let A be an array of n numbers. You are given a number x. Design an algorithm that finds out distinct indices $i, j, k \in \{1, 2, \ldots, n\}$ such that $A[i] + A[j] + A[k] = x$. Your algorithm must run in $O(n^2)$-time.

10. Let A and B be two arrays of n numbers each having n distinct numbers. Define

$$Pred(a) = \max_{b \in B}\{b < a\}, a \in A$$

Informally, for every $a \in A$, the $Pred(a)$ is the largest number $b \in B$ such that $b < a$. Design an $O(n \log n)$-time algorithm to compute $Pred(a)$ for all $a \in A$.

11. You are given two sorted arrays $A[]$ and $B[]$ of positive integers. The sizes of the arrays are not given. Accessing any index beyond the last element of the arrays returns -1. The elements in each arrays are distinct but the two arrays may have common elements. An intersection point between two arrays is an element that is common to both, i.e. p be an intersection point if there is i and j such that $A[i] = B[j] = p$.

    Given an integer x design an algorithm (in pseudocode) to check if x is an intersection point of A and B. Your algorithm must run in time asymptotically faster than linear in the maximum size of the two arrays.

12. Suppose you are given two sets $\{p_1, \ldots, p_n\}$ and $\{q_1, \ldots, q_n\}$ of n points on the unit circle. For every $i \in [n]$, connect p_i to the point q_i. Design a divide and conquer algorithm to determine how many pairs of these line segments intersect in $O(n \log^2 n)$-time.