Tutorial 2, Design and Analysis of Algorithms, 2020

1. The sets A and B have n elements each given in the form of sorted arrays. Design $O(n)$ algorithms to compute $A \cup B$ and $A \cap B$.

2. Consider two sets A and B, each having n integers in the range from 0 to $10n$. We wish to compute the Cartesian sum of A and B, defined by
$C = \{x + y \mid x \in A \land y \in B\}$
Note that the integers in C are in the range from 0 to $20n$. We want to find the elements of C and the number of times each element of C is realized as a sum of elements in A and B. Show how to solve the problem in $O(n \log n)$ time.

3. Arrange the following algorithms in increasing order of time complexity: *Karatsuba's Divide and Conquer Integer Multiplication Algorithm*, *Iterative Matrix Multiplication Algorithm*, *Strassen's Divide and Conquer Matrix Multiplication Algorithm*. Give proper reasons for your answer.

4. Find 67698×8967 using *Karatsuba's Divide and Conquer Integer Multiplication Algorithm* showing the computations in a divide and conquer graph.

5. Professor Caesar wishes to develop a matrix-multiplication algorithm that is asymptotically faster than Strassen's algorithm. His algorithm will use the divide-and-conquer method, dividing each matrix into pieces of size $\frac{n}{4} \times \frac{n}{4}$, and the divide and combine steps together will take $\Theta(n^2)$ time. He needs to determine how many subproblems his algorithm has to create in order to beat Strassen's algorithm. If his algorithm creates a subproblems, then the recurrence for the running time $T(n)$ becomes $T(n) = aT\left(\frac{n}{4}\right) + \Theta(n^2)$. What is the largest integer value of a for which Professor Caesar's algorithm would be asymptotically faster than Strassen's algorithm?

6. (a) Find the number of nodes in the divide and conquer graph for computing FFT of a vector of length n (for simplicity you can assume n to be a power of 2).

   (b) Now find time complexity of the FFT algorithm by only considering the structure of the divide and conquer graph (without solving any recursion).

7. Find 1234×4321 using the FFT algorithm showing its divide and conquer graphs.

8. (a) Describe the generalization of the FFT procedure to the case in which n is a power of 3 (using three subproblems). Give a recurrence for the running time, and solve the recurrence.

   (b) Find 97×68 using the above algorithm showing its divide and conquer graphs.

9. Consider 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 $1 \le i \le n$ and $1 \le j \le n$; the nodes (i, j) and (k, l) are joined by an edge if and only if $|i - k| + |j - l| = 1$.) Each node v of G 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 G 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 an $n \times n$ grid graph G, 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 G using only $O(n)$ *probes* to the nodes of G. Give a proof of correctness of your algorithm and also prove its time complexity.

10. In the *Josephus Problem*, we start with n people numbered 1 to n around a circle, and we eliminate every *second* remaining person until only one survives. For example, the elimination order for $n = 10$ is $2, 4, 6, 8, 10, 3, 7, 1, 9$, so 5 survives. The problem is to determine the survivor's number, $J(n)$ (in the above example, we have $J(10) = 5$). Design a linear time complexity algorithm for computing $J(n)$.