ECSE210L: Design and Analysis of Algorithms

Tutorial 4 (Week 4: January, 27 - 31, 2020)

Instructors: ***Shakti Sharma and Raghunath Reddy M***

(**Q1**) Let A be a $n \times n$ matrix with distinct elements and the elements are sorted in both row-wise and column-wise. Note that matrix A has $N = n^2$ elements. You are given an element *key*. The goal is to verify whether element *key* is in A or not.

It is easy to see an $O(N)$-time algorithm which compares each element in A with *key*. Design a better algorithm using Divide and Conquer method.

(**Q2**) Let $A = [a_1, a_2, \ldots, a_n]$ be an array of n elements. Define a left-circular shift operation on A as follows: shift every element in A to one position to the left in the circular fashion i.e., after one left-circular shift, $A = [a_2, a_3, \ldots, a_n, a_1]$.

Now, you are given an array A of n distinct integers such that one can arrange the elements in A in increasing order by performing t left-circular shifts on A for some t with $0 \leq t \leq n-1$. Your job is find the exact value of t.

Examples:

(1) Let $A = [54, 76, 99, 15, 37, 44, 49]$. The elements is A can be arranged in increasing order by performing three left-circular shifts.
(2) Let $A = [11, 22, 33, 44, 1]$. We need 4 left-circular shifts to obtain the increasing order of elements in A.

One can easily give an $O(n)$-time algorithm. Design a $O(\log n)$-time algorithm ?

Hint: Find the position of the largest element in A.

(**Q3**) Consider the following two matrices: $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ and $B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$

The multiplication of A and B is given as: $AB = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix}$

We now use this to find the multiplication of two matrices of size $n \times n$ each.

Let A and B be two matrices of size $n \times n$ each. Assume that n is a power of 2 (this is not a strict assumption). Now divide each matrix, A and B, into four quarters as shown below:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

where A_{ij} and B_{ij} are the matrices of size $n/2 \times n/2$.

Then, the multiplication of A and B is given as: $AB = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix}$

One can compute the multiplication of A and B by recursively sub-dividing each sub-matrix into smaller matrices till the size reaches to sufficiently small.

What is the time complexity of the above algorithm ?