Agenda for today's live Session

Application of Divide and Conquer

Selection Procedure

Problem Statement :

Input : An array of n elements and the value of k

Output : To return kth smallest element

Selection Procedure Algorithm :

Value of m is computed with the help of Partition algorithm that we have studied earlier in Quicksort Algorithm concept.

```
m = Partition(arr,p,q)      O(n)

SelectionProcedure(arr,p,q,k){

        if(k == m){

                return arr[m];                  O(1)

        }

        else if(k < m){

                SelectionProcedure(arr,p,m-1,k);      T(m-p)

        }

        else if(k > m){

                SelectionProcedure(arr,m+1,q,k);      T(q-m)

        }

}
```

Recurrence Relation of Selection Procedure :

T(n) = O(n) + O(1) + T(m-p) or T(q-m)

Note : Here, either we go towards left part or right side in an array and this is the reason I mentioned or between T(m-p) and T(q-m).

Best case scenario :

T(n) = O(n) + T(n/2)

Using master's theorem

n^log_b a = n^log_2 1 = n^0 = 1

f(n) = n

Overall time complexity = O(n)

Worst case scenario :

T(n) = O(n) + T(n-1)

Using substitution method,

T(n) = O(n^2)