

Name : Pathak Viraj Hemangkumar  
ID : 1002195291  
Hands - On 6

### 3. Mathematically derive the average runtime complexity of the non-random pivot version of quicksort

Solution :

In the non-random pivot version of Quicksort, we typically choose the pivot element deterministically. One common approach is to choose the pivot as the first, last, or middle element of the array. For the sake of simplicity, let's assume we choose the pivot as the first element of the array.

The average case analysis of Quicksort depends on the partitioning process. In each partitioning step, we partition the array into two sub-arrays such that elements in one sub-array are less than or equal to the pivot, and elements in the other sub-array are greater than the pivot. If the partitioning is balanced, then Quicksort will perform well.

Let  $T(n)$  denote the average runtime complexity of Quicksort for an input array of size  $n$ . In each partitioning step, we have to scan through all  $n$  elements of the array once to partition it.

After partitioning, let's say the pivot splits the array into two sub-arrays of sizes  $k$  and  $(n-k-1)$ , where  $k$  is the number of elements less than the pivot.

The recurrence relation for the average case time complexity  $T(n)$  can be expressed as follows:

$$T(n) = \text{average time to partition} + T(k) + T(n-k-1)$$

The "average time to partition" refers to the time taken to partition the array, which is linear in  $n$ .

If we can prove that on average, the partitioning process divides the array into two roughly equal parts, we can establish that  $k$  is approximately  $n/2$ .

Thus, the recurrence relation becomes:

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

Now, we need to solve this recurrence relation.

By the Master theorem, the solution to this recurrence relation is:

$$T(n) = O(n \log n)$$

This is because in each recursive call, the array size is halved, and we have a linear partitioning step which gives us the  $O(n \log n)$  average runtime complexity for Quicksort with the non-random pivot strategy.

Therefore, the average runtime complexity of the non-random pivot version of Quicksort is  $O(n \log n)$ .

