Role Play Activity

Subject: Design and Analysis of Algorithms

Date: 28/1/2020

Topic: Quick Sort

Participants: 18911A0563, 572, 575, 581, 586, 569, 590, 5B3, 5A9, 19915A0509

Content:

Divide: Rearrange the elements and split arrays into two sub-arrays and an element in between search that each element in left sub-array is less than or equal to the average element and each element in the right sub-array is larger than the middle element.

Conquer: Recursively, sort two sub-arrays.

Combine: Combine the already sorted array.

Algorithm:

1. QUICKSORT (array A, int m, int n)
2. 1 if (n > m)
3. 2 then
4. 3 i ← a random index from [m,n]
5. 4 swap A[i] with A[m]
6. 5 o ← PARTITION (A, m, n)
7. 6 QUICKSORT (A, m, o - 1)
8. 7 QUICKSORT (A, o + 1, n)

Partition Algorithm:

Partition algorithm rearranges the sub-arrays in a place.

1. PARTITION (array A, int m, int n)
2. 1 x ← A[m]
3. 2 o ← m
4. 3 for p ← m + 1 to n
5. 4 do if (A[p] < x)
6. 5 then o ← o + 1
7. 6 swap A[o] with A[p]
8. 7 swap A[m] with A[o]
9. 8 return o
Figure: shows the execution trace partition algorithm