

Algorithms and Complexity – CS700

Practice Lab Assignment - 01

Problem Statement: Quicksort is an algorithm based on divide and conquer approach in which an array that contains a set of elements is split into subarrays and these sub-arrays are recursively called to sort the elements.

So far, you have seen running time complexity of the Quicksort algorithm expressed in the big-Oh notation. In order to put the theoretical study of the algorithm to practical tests, you take this sorting algorithm in this assignment and measure its actual running time along with the space on inputs of various sizes. Let n (input size) denotes the size of an array A that contains n elements (non-negative integers) in unsorted order. We want to sort this array. Minimum value for n must be one lakh, i.e., the A contains at least 100000 elements.

The performance of quick sort is very sensitive to the initial distribution of the array A and also to the choice of the pivot. You need to work with the following ways of choosing the pivot for partitioning the array while attempting to sort the array.

- FIRST Position: Choose the first element $A[0]$ as the pivot.
- RANDOM Position: Choose $A[r]$ as the pivot for a random $r \in \{0, 1, 2, \dots, n-1\}$.
- MEDIAN Position: Choose $A[r/2]$ as the pivot where $r=n/2$.
