Praveen Anand
1002219407

Hands-on-6

3) Mathematically derive the average runtime Complexity of the non-random Pivot version of quicksort.

The recurrence for quicksort with a non-Pivot is

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

where k is the number of elements less than the Pivot.

Best case:-

When the Pivot always divides the array into two equal halves, ie, $k = n/2$ the recurrence relation becomes

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

Solving this we get time Complexity $O(n \log n)$

Worst Case: When Pivot always the smallest or largest element, the recurrence becomes.

$T(n) = T(n-1) + O(n)$ Solving this we get $O(n^2)$.

Average Case:- On Average, the Pivot will divide the arrays into two Subarrays that are approximately equal in Size. The expected Value of k is $n/2$, and the recurrence is:-

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

This recurrence solve to $O(n \log n)$, meaning the average time Complexity is also $O(n \log n)$.

Non random Pivot quick sort has best & average case complexities of $O(n \log n)$ and worst-case complexity of $O(n^2)$.