Hands-on-6

Pallavi Chowdary Gogineni
1002221774

Mathematically derive the average runtime complexity of the non-Random Pivot version of quick sort

Let $T(n)$ be the average time complexity for quick sort on an input size n.

Here Partition step takes $O(n)$ times as it needs to compare each element to the Pivot.

After Partitioning, we recursively sort two subarrays size approximately $n/2$ for each.

Therefore recurrence relation $T(n) = 2T(n/2) + O(n)$

using master theorem

$a = 2$ (number of recursive calls)

$b = 2$ (size reduction factor in each recursive call)

$f(n) = O(n)$ (work done outside of recursive call)

$f(n) = O(n)$

$= \Theta(n \log_b(a)) = \Theta(n \log_2(2)) = \Theta(n)$

$\therefore$ Therefore $T(n) = \Theta(n \log n)$

Thus the average runtime complexity of the non-Random Pivot version of quicksort is $O(n \log n)$