Hands On - 6

3(A)

So for Average Case Derivation

Step 1: Partitioning work

→ Select a pivot

→ perform Comparisions to divide the array into two parts

This takes $O(n)$ time

So after partioning array divides into two parts

size i and $n-i-1$ (i is left of pivot elements)

Step 2: Recursive work

So avg case we sum over all possible size of the left Subarray i, weighted by the probability of selecting that pivot

$$T(n) = O(n) + \frac{1}{n} \sum_{i=0}^{n-1} \left(T(i) + \underbrace{T(n-i-1)}_{\text{almost equal}} \right)$$

So we simply this to

$$T(n) = O(n) + \frac{2}{n} \sum_{i=0}^{n-1} T(i)$$

Now apply masters theorm.

$$T(n) = aT\left(\frac{n}{b}\right) + O(n^d)$$

For Quick Sort

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

$a = 2$ 2 recursive calls

$b = 2$ For each recursive call on subproblem of size $\frac{n}{2}$

$d = 1$ $O(n)$

compare n^d with $n^{\log_b a}$

$$n^{\log_2 2} = n^1$$

so here $n^d = n^1$

Case 2: $d = \log_b a$

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

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

Time complexity of Quick Sort