Hands on 6 (N0 3)

1. Mathematically derive the average runtime complexity of the non-randomized pivot version of quicksort.
   - Solution:
     Here,

⇨ To derive the average runtime complexity of the non-randomized pivot of quicksort we must follow the below steps:

⇨ **Recurrence Relation:**

- At each step, Quicksort partitions the array around a pivot.

- On average, it splits the array into two subarrays of approximately equal size.
- The recurrence relation is:

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

- The O(n) term represents partitioning.

⇨ **Expanding the Recurrence:**

- Expand for multiple levels:

$$T(n) = 2T(\frac{n}{2}) + cn$$

$$= 2(2T(\frac{n}{4}) + c(\frac{n}{2})) + cn$$

$$= 4T(\frac{n}{4}) + 2cn + cn$$

$$= 8T\left(\frac{n}{8}\right) + 3cn$$

- Continue expanding until $T(1)$, which takes constant time.

⇨ **Depth of Recursion Tree:**

- The recursion depth is determined by how many times n can be divided by 2 until it reaches 1.
- This occurs at depth $\log_2 n$ $meaning\ O(logn)$ levels.

- **Total Work Per Level:**

  - Each level of recursion does $O(n)$ work.
  - There are $O(logn)$ levels.

- Final **Complexity:**

  - Summing across all levels: $O(n) \times O(logn) = O(nlogn)$
  - Thus, the **average runtime complexity** of **Quicksort** is $\boldsymbol{O(n\ log\ n)}$.