2)Best,Worst and Average Time:

A graph with different colored lines

Description automatically generated

3) **Deriving Average Runtime Complexity:**

The average runtime complexity of non-random pivot quicksort can be analyzed using the **Master Theorem** for divide-and-conquer algorithms.

**Master Theorem:**

* T(n) = a \* T(n/b) + f(n)
* Where:
  + T(n) is the time complexity for input size n.
  + a is the number of subproblems created (2 for quicksort - left and right sub-arrays).
  + b is the size factor by which the subproblems are smaller (2 for quicksort - sub-arrays are half the size).
  + f(n) is the work done at each level (proportional to n for partitioning).

**Applying the Master Theorem:**

* a = 2
* b = 2
* f(n) = O(n)
* The Master Theorem states that for a = b^d and f(n) = O(n^(d-1)), the complexity is O(n^d log n).

In this case, a = b^1 and f(n) = O(n^0), which doesn't satisfy any of the cases in the Master Theorem directly. However, for f(n) = O(n^(d-ε)) where ε > 0 (in this case, ε = 1), the complexity becomes **O(n log n)**.

Therefore, the average runtime complexity of non-random pivot quicksort is **O(n log n)** due to balanced partitioning on average over many random inputs. However, it is crucial to remember that the worst-case scenario remains O(n^2) with specifically crafted inputs.