**BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY**
**Department of Computer Science and Engineering**

**Course:** CSE204 (Data Structure and Algorithm I Sessional)

**Assignment 7**

**Assignment name:** SORTING ALGORITHMS.

**Name:** Hasan Masum

**Student ID:** 1805052

## Objective

In this assignment, after implementing Quick and Merge sort, their performance are compared.

## Language Use: C++

## Data Table:

| Input Order | n= / Sorting Algorithm | 10 | 100 | 1000 | 10000 | 100000 | 100000 |
|---|---|---|---|---|---|---|---|
| Ascending | Merge | 0.001 | 0.001 | 0.001 | 0.002 | 0.010 | 0.115 |
| | Quick | 0.001 | 0.001 | 0.0042 | 0.194 | 18.92 | 1927.38 |
| Descending | Merge | 0.001 | 0.001 | 0.001 | 0.002 | 0.011 | 0.112 |
| | Quick | 0.001 | 0.001 | 0.003 | 0.152 | 14.41 | 1325.369 |
| Random | Merge | 0.001 | 0.001 | 0.001 | 0.002 | 0.017 | 0.19 |
| | Quick | 0.001 | 0.001 | 0.001 | 0.003 | 0.02 | 0.23 |

Table: ***Average time*** for sorting ***n*** integers in different **input orders**

## Input Size & Runtime Graphs:

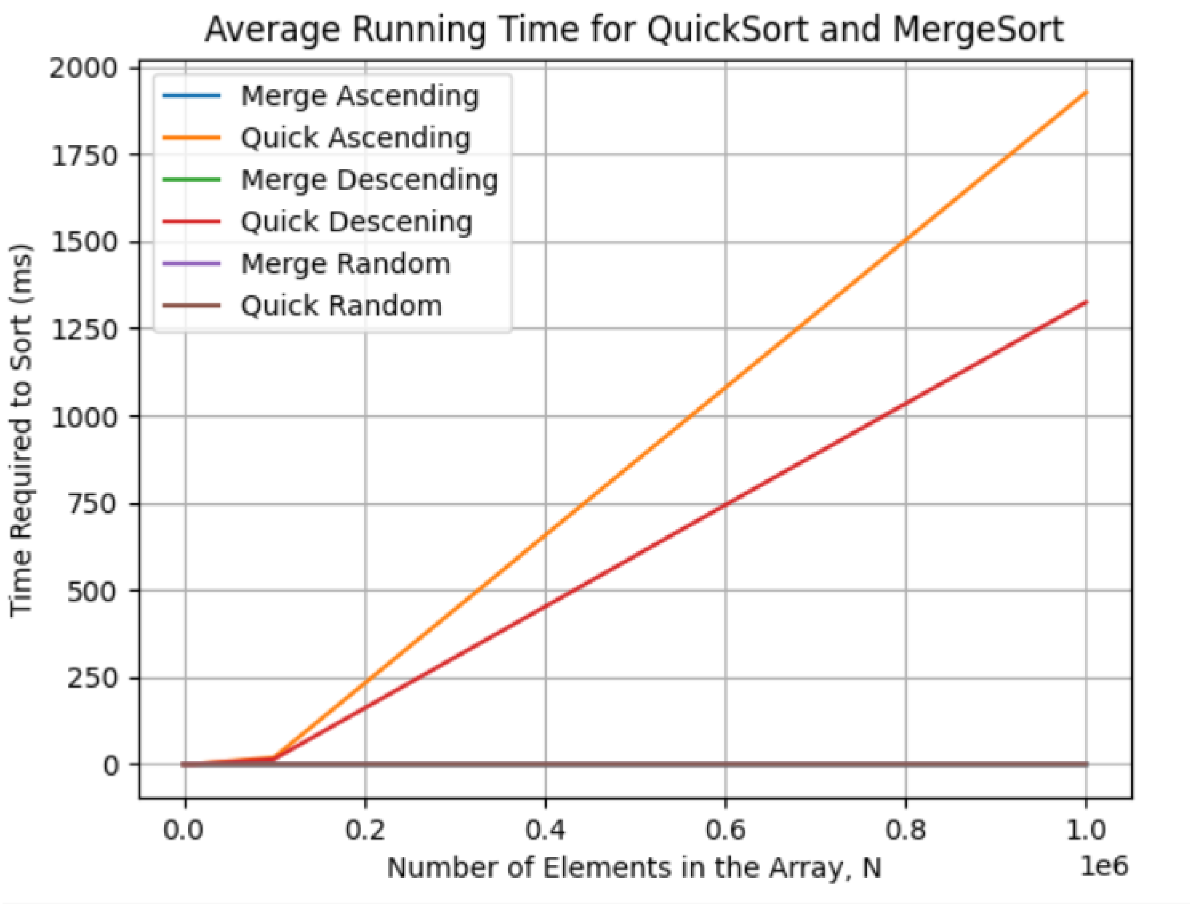



Fig: Runtime Vs Number of input graph for all test case for both Merge and Quick Sort

## Complexity analysis:

For complexity analysis we need to focus on following:

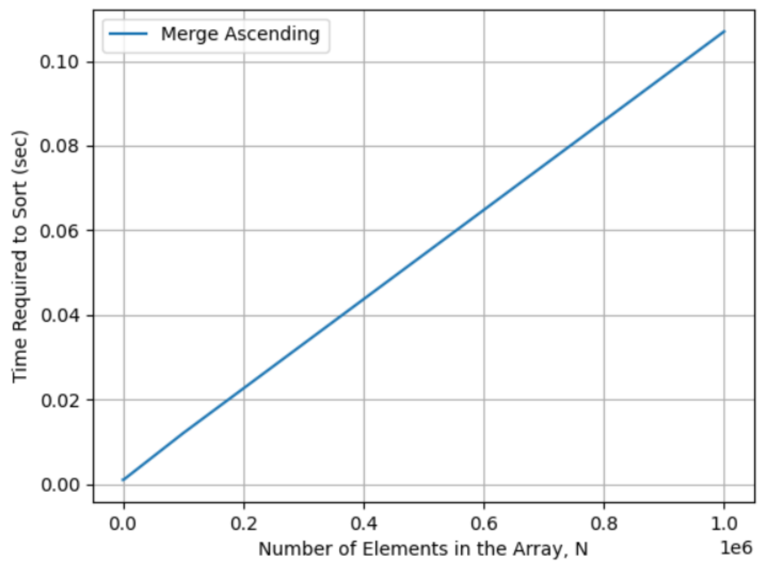
- How long the algorithm takes in terms of input size.
- How fast the graph grows with the input size.
- Finally we have to determine a complexity class.

Again we know runtime is machine dependent and runtime is different from machine to machine. But it only varies by a constant. We can eliminate this constant by taking ratio of runtime for different input size.

Let's assume T(n) is the runtime we got from the test for input size n. if O(n) is the complexity then we can say that T(n) = k*O(n). We can eliminate the machine dependent constant k by taking ration of the runtimes.

## Order: Ascending

**Merge:**



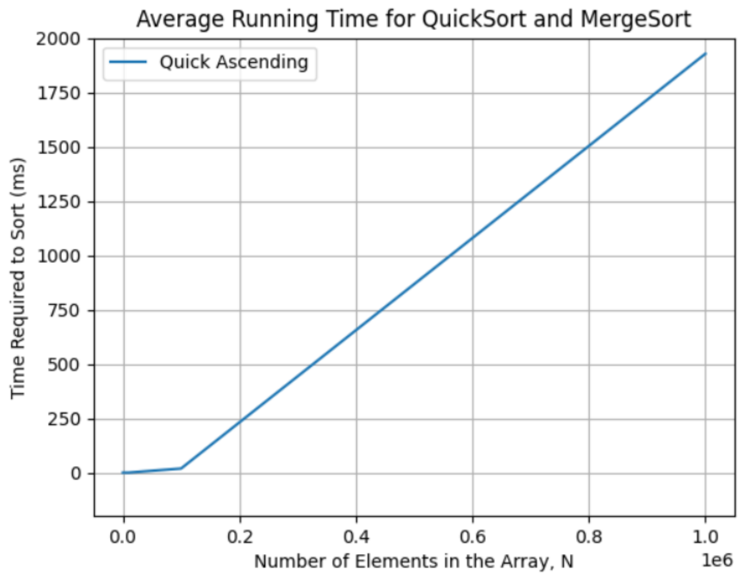

Here,

$$\frac{T(10^6)}{T(10^5)} = \frac{0.115}{0.010} = 11.5$$

For O(nlog(n))

$$\frac{10^6 \log(10^6)}{10^5 \log(10^5)} = 12$$

**Quick:**




Here,

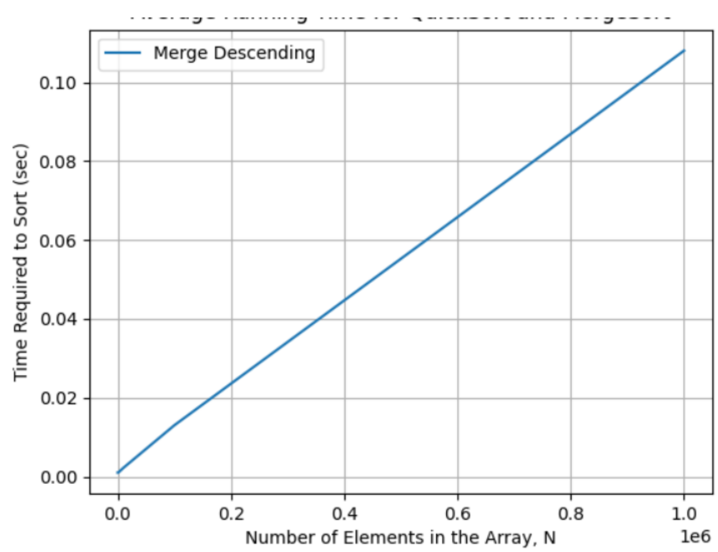
$$\frac{T(10^6)}{T(10^5)} = \frac{1927.38}{18.92} = 101.87$$

For O(n$^2$)

$$\frac{10^6 * 10^6}{10^5 * 10^5} = 100$$

## Order: Descending

### Merge:



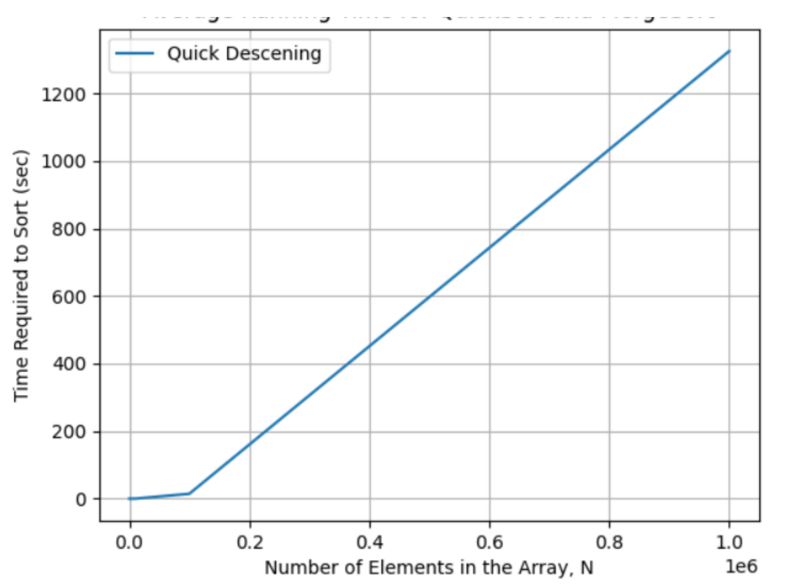

Here,

$$\frac{T(10^6)}{T(10^5)} = \frac{0.112}{0.011} = 10.18$$

For O(nlog(n))

$$\frac{10^6 \log(10^6)}{10^5 \log(10^5)} = 12$$

**Quick:**




Here,

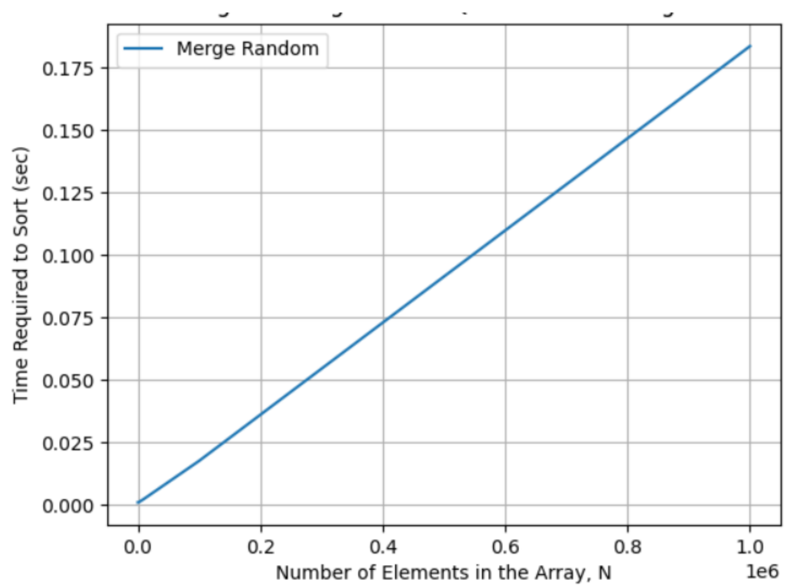
$$\frac{T(10^6)}{T(10^5)} = \frac{1325.369}{14.41} = 91.98$$

For O(n$^2$)

$$\frac{10^6 * 10^6}{10^5 * 10^5} = 100$$

## Order: Random

### Merge:



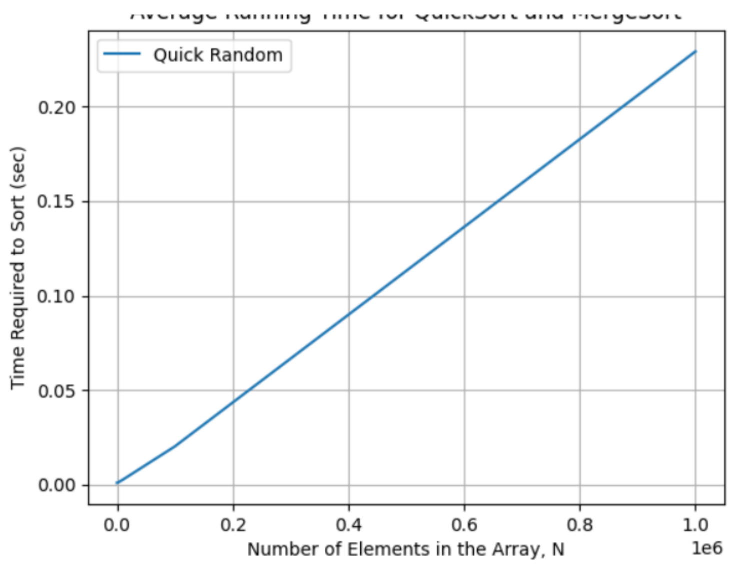

Here,

$$\frac{T(10^6)}{T(10^5)} = \frac{0.19}{0.017} = 11.18$$

For O(nlog(n))

$$\frac{10^6 \log(10^6)}{10^5 \log(10^5)} = 12$$

**Quick:**




Here,

$$\frac{T(10^6)}{T(10^5)} = \frac{0.23}{0.02} = 11.5$$

For O(nlog(n))

$$\frac{10^6 \log(10^6)}{10^5 \log(10^5)} = 12$$