# Report : Project III

## PART 3.2:

**Selection Sort:**

**Initial Vector:**
42 73 89 14 51 3 37
**Number of Comparisons in Selection Sort:** 21
**Number of Assignments in Selection Sort:** 6
**Took**: 69567

**Explanation**:

- For counting the number of comparisons, I am incrementing the counter every time I compare an array value with the maxPos array value.  So basically it is equal to the number of times we run the while loop for finding the largest item.
- For counting the number of assignments, I am updating the counter everytime we do a swap where we assume the assignment to be equal to one swap.

**Bubble Sort**:
**Number of Comparisons in Bubble Sort**: 36
**Number of Assignments in Bubble Sort**: 14
**Took**: 84994

**Explanation**:

- For counting the number of comparisons, I am incrementing the counter every time we do a comparison by calling the compare method.
- For counting the number of assignments, I am updating the counter everytime we do a swap where we assume the assignment to be equal to one swap.

**Merge Sort:**
**Number of Comparisons in Merge Sort**: 13
**Number of Assignments in Merge Sort:** 20
**Took:** 72069

**Explanation**:

- For counting the number of comparisons, I update the counter when  the comparison is done on the sorted halves for merging.
- For counting the number of assignments, I update the counter, everytime, we do an assignment in the array by comparing sorted halves.

**Quick Sort:**
**Number of Comparisons in Quicksort**: 10
**Number of Assignments in Quick Sort**: 9
**Took**: 60652

**Explanation**:

- For counting the number of comparisons, I update the counter everytime we do a comparison for putting smaller values  on one side of array and larger values on the other side of array
- For counting the number of assignments, I update the counter, when we do the assignment of values from the smaller vector first, (for each value assigned, we update counter) then pivot and then the larger vector (for each value assigned, we update counter) .

## PART 3.3:

**Time:**

| | 100 Random (Avg) | 1000 Random (Avg) | 10,000 Random (Avg) | 100,000 Random (Avg) |
|---|---|---|---|---|
| **Selection Sort** | **Runtime**: 220100<br>**Comparison**: 4950<br>**Assignment**: 99 | **Runtime**: 7232441<br>**Comparison**: 499500<br>**Assignment**:999 | **Runtime**: 873508921<br>**Comparison**: 49995000<br>**Assignment**:9999 | **Runtime**: 1244549036<br>**Comparison**: 704982704<br>**Assignment**:99999 |
| **Bubble Sort** | **Runtime**: 888091<br>**Comparison**:9801<br>**Assignment**: 2458 | **Runtime**: 31574497<br>**Comparison**: 998001<br>**Assignment**: 245881 | **Runtime**: 134269589<br>**Comparison**: 99980001<br>**Assignment**: 15148252 | **Runtime**: Too Long to calculate<br>**Comparison**: 999800001<br>**Assignment**: Varied |
| **Merge Sort** | **Runtime**: 221421<br>**Comparison**:542 | **Runtime**: 997363<br>**Comparison**: | **Runtime**: 22748432<br>**Comparison**: | **Runtime**: 274444399 |

| | | | | |
|---|---|---|---|---|
| | **Assignment**:672 | 8697<br>**Assignment**: 9976 | 120444<br>**Assignment**:133616 | **Comparison**: 4950<br>**Assignment**: 99 |
| **Quick Sort** | **Runtime**: 807844<br>**Comparison**:637<br>**Assignment**: 150 | **Runtime**: 2124742<br>**Comparison**: 10742<br>**Assignment**: 1493 | **Runtime**: 42814346<br>**Comparison**: 155023<br>**Assignment**: 15061 | **Runtime**: 226747426<br>**Comparison**: 1536410<br>**Assignment**: 1668928 |

(Values in this table are derived from averages of the tables below analysis)

## Analysis :

**Fastest and slowest performance at specific vector sizes :**

- For 100 Random numbers, running time is least for selection sort and highest for bubble sort
- For 1000 Random numbers, running time is least for merge sort and highest for bubble sort
- For 10000 Random numbers, running time is least for merge sort and highest for selection sort
- For 100000  Random numbers, running time is least for quick sort and highest for bubble sort

**General comparison of running time of sorts in all vector sizes  :**

- Based on all of the above vector sizes, merge sort tends to be the fastest with the lowest running time for two vector sizes
- Bubble sort tends to be the slowest with the highest running time for 3 different vector sizes
- Merge and Quicksort tend to be faster than bubble and selection sort based on the data above

**Similarity/Difference in performance between algorithms at specific vector size**

- The algorithms tend to perform similarly with similar running times at 100 Random numbers.
- They show a huge difference in performance when the vector size is increased to a 100 as bubble and selection sort tend to become way more slower than merge and quicksort.

**Selection Algorithm:**

|  | 1st try | 2nd try | 3rd try | 4 try | 5 try | Avg |
|---|---|---|---|---|---|---|
| 100 values | RT:156926<br>Comparison:4950<br>Assignment: 99 | RT:210333<br>Comparison: 4950<br>Assignment: 99 | RT:239321<br>Comparison:4950<br>Assignment: 99 | RT:239321<br>Comparison: 4950<br>Assignment: 99 | RT:254602<br>Comparison:4950<br>Assignment: 99 | RT:220100<br>Comparison: 4950<br>Assignment: 99 |
| 1000 values | RT:7185079<br>Comparison:499500<br>Assignment: 999 | RT:7092186<br>Comparison: 499500<br>Assignment: 999 | RT:6896102<br>Comparison:499500<br>Assignment: 999 | RT:7031998<br>Comparison: 499500<br>Assignment: 999 | RT:7956840<br>Comparison:499500<br>Assignment: 999 | RT:7232441<br>Comparison: 499500<br>Assignment: 999 |
| 10,000 values | RT:738194197<br>Comparison:49995000<br>Assignment: 9999 | RT:906367491<br>Comparison: 49995000<br>Assignment: 9999 | RT:883391563<br>Comparison:49995000<br>Assignment: 9999 | RT:924874172<br>Comparison: 49995000<br>Assignment: 9999: | RT:914717183<br>Comparison:49995000<br>Assignment: 9999 | RT:873508921<br>Comparison: 49995000<br>Assignment: 9999 |
| 100,000 values | RT:999038365<br>Comparison:704982704<br>Assignment: 99999 | RT:2095338206<br>Comparison: 704982704<br>Assignment: 99999 | RT:365607588<br>Comparison:704982704<br>Assignment: 99999 | RT:2044907795<br>Comparison: 704982704<br>Assignment: 99999 | RT:717853227<br>Comparison:704982704<br>Assignment: 99999 | RT:1244549036<br>Comparison: 704982704<br>Assignment: 99999 |

**Bubble Sort Algorithm:**

|  | 1st try | 2nd try | 3rd try | 4 try | 5 try | Avg |
|---|---|---|---|---|---|---|
| 100 values | RT:651224<br>Comparison:9801<br>Assignment: 2183 | RT:359297<br>Comparison:9801<br>Assignment: 2707 | RT:2842719<br>Comparison: 9801<br>Assignment: 2268 | RT:239343<br>Comparison:9801<br>Assignment: 2789 | RT:347874<br>Comparison:9801<br>Assignment: 2345 | RT:888091<br>Comparison:9801<br>Assignment: 2458 |
| 1000 values | RT:31106603<br>Comparison:998001<br>Assignment: 250375 | RT:65390493<br>Comparison:998001<br>Assignment: 245083 | RT:21899520<br>Comparison: 998001<br>Assignment: 245165 | RT:22208979<br>Comparison:998001<br>Assignment: 244976 | RT:17266892<br>Comparison:998001<br>Assignme | RT:31574497<br>Comparison:998001<br>Assignment: 245881 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | nt:243808 | |
| 10,000 values | RT:125817205<br>Comparison:99980001<br>Assignment: 24894251 | RT:272516822<br>Comparison:99980001<br>Assignment: 25238575 | RT:210800715<br>Comparison: 99980001<br>Assignment: 25107685 | RT:31106603<br>Comparison:99980001<br>Assignment: 250375 | RT:31106603<br>Comparison:99980001<br>Assignment: 250375 | RT:134269589<br>Comparison:99980001<br>Assignment: 15148252 |
| 100,000 values | 511730736<br>Comparison:999800001 | - | - | - | - | - |

**QuickSort Algorithm:**

| | 1st try | 2nd try | 3rd try | 4 try | 5 try | Avg |
|---|---|---|---|---|---|---|
| 100 values | RT:304132<br>Comparison:754<br>Assignment: 153 | RT:2962012<br>Comparison:677<br>Assignment: 150 | RT:308541<br>Comparison:564<br>Assignment: 156 | RT:195865<br>Comparison:614<br>Assignment: 147 | RT:268671<br>Comparison: 576<br>Assignment: 147 | RT:807844<br>Comparison:637<br>Assignment: 150 |
| 1000 values | RT:2253017<br>Comparison:10842<br>Assignment: 1467 | RT:2516090<br>Comparison: 11020<br>Assignment: 1518 | RT:1745447<br>Comparison:10068<br>Assignment: 1476 | RT:1719795<br>Comparison:10842<br>Assignment: 1503 | RT:2389364<br>Comparison:10939<br>Assignment: 1500 | RT:2124742<br>Comparison:10742<br>Assignment: 1493 |
| 10,000 values | RT:18505595<br>Comparison: 146569<br>Assignment: 15045 | RT:76823984<br>Comparison:160316<br>Assignment: 14964 | RT:20078704<br>Comparison: 151893<br>Assignment: 15087 | RT:61428099<br>Comparison: 161443<br>Assignment: 15030 | RT:37235350<br>Comparison: 154897<br>Assignment: 15183 | RT:42814346<br>Comparison: 155023<br>Assignment: 15061 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 100,000 values | RT:496872196<br>Comparison:2100587<br>Assignment: 151041 | RT:461623609<br>Comparison:1919700<br>Assignment: 151146 | RT:393121032<br>Comparison:2072250<br>Assignment: 150606 | RT:422726048<br>Comparison:2048880<br>Assignment:150738 | RT:551018327<br>Comparison:2033193<br>Assignment: 150807 | RT:465072242<br>Comparison:2034922<br>Assignment: 150867 |

**Merge Algorithm:**

| | 1st try | 2nd try | 3rd try | 4 try | 5 try | Avg |
|---|---|---|---|---|---|---|
| 100 values | RT:210010<br>Comparison: 540<br>Assignment: 672 | RT:143480<br>Comparison:531<br>Assignment: 672 | RT:415403<br>Comparison: 543<br>Assignment: 672 | RT:154097<br>Comparison:547<br>Assignment:672 | RT:184117<br>Comparison:546<br>Assignment:672 | RT:221421<br>Comparison: 542<br>Assignment: 672 |
| 1000 values | RT:832452<br>Comparison: 8696<br>Assignment: 9976 | RT:1007279<br>Comparison:8682<br>Assignment: 9976 | RT:1118916<br>Comparison: 8702<br>Assignment: 9976 | RT:1257287<br>Comparison:8716<br>Assignment:9976 | RT:770883<br>Comparison:8692<br>Assignment:9976 | RT:997363<br>Comparison: 8697<br>Assignment: 9976 |
| 10,000 values | RT:7838287<br>Comparison: 120364<br>Assignment: 133616 | RT:67444069<br>Comparison:120451<br>Assignment: 133616 | RT:11166155<br>Comparison: 120462<br>Assignment: 133616 | RT:14877369<br>Comparison:120477<br>Assignment:133616 | RT:12416282<br>Comparison:120469<br>Assignment:133616 | RT:22748432<br>Comparison: 120444<br>Assignment: 133616 |
| 100,000 values | RT:223365891<br>Comparison: 1536426<br>Assignment: 1668928 | RT:285422488<br>Comparison:1536432<br>Assignment: 1668928 | RT:203088066<br>Comparison: 1536383<br>Assignment: 1668928 | RT:199539764<br>Comparison:1536230<br>Assignment:1668928 | RT:222320922<br>Comparison:1536580<br>Assignment:1668928 | RT:226747426<br>Comparison: 1536410<br>Assignment: 1668928 |