DAA Project Report
Sorting Algorithms

- Language used for the project - Javascript, HTML and CSS
- Libraries used - canvas.js for implementing the bar graph - https://canvasjs.com/html5-javascript-column-chart/
  Added the script using cdn link
- References - geeksforgeeks - https://www.geeksforgeeks.org/
- Run time calculations - Used the inbuilt function window.performance.now() to calculate the run times for the algorithm

Global variables used

1. input - This will contain the input array, once the user selects the random array or input array option. We then set the input array variable accordingly
2. data - This contains the output array, it will contain multiple outputs everytime we generate a new input array its corresponding output will be pushed into data. Each element in the array is again the array of objects. The object will have a key which will be the type of sorting algorithm and the value will be an object with sortedArray and time. For example, we have generated two input arrays, then the data will be
   ```
   data = [
           [
           {linear;{sortedArray:[], time:number}},
           {bubble;{sortedArray:[], time:number}},
           {selection;{sortedArray:[], time:number}},
           {merge;{sortedArray:[], time:number}},
           {heap;{sortedArray:[], time:number}},
           {quick;{sortedArray:[], time:number}},
           {threeWayQuick;{sortedArray:[], time:number}}
           ],
           [
           {linear;{sortedArray:[], time:number}},
           {bubble;{sortedArray:[], time:number}},
           {selection;{sortedArray:[], time:number}},
           {merge;{sortedArray:[], time:number}},
           {heap;{sortedArray:[], time:number}},
           {quick;{sortedArray:[], time:number}},
           {threeWayQuick;{sortedArray:[], time:number}}
           ]
          ]
   ```
3. arrLength - this will contain length of all the input arrays
4. checkedSortingAlgoArray - contains array of selected algorithms through the checkbox

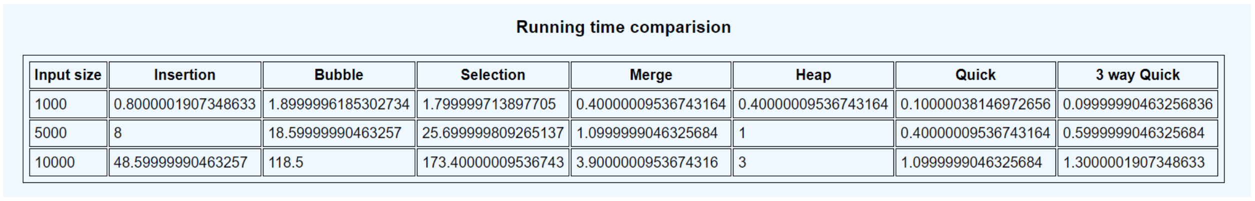
Main Functions used

1. setInput() - This function is called when we click on the time comparison button, this function will generate the random array or create an array from the user input based the option selected by the user. This input array is assigned to the global variable input which is used throughout by all the sorting algorithm functions
2. displayTableAndGraph() - This function is called after running all the sorting algorithms. It will create the table and graph for all the outputs present in the data variable
3. insertionSort() - code for insertion sort
4. selectionSort() - code for selection sort
5. bubbleSort() - code for bubble sort
6. mergeSort(), mergeSortAlgo(), merge() - functions used for merge sort
7. quickSort(), quickSortAlgo(), partition() - functions used for quick sort and we have used pivot for the last element
8. threeWayQuickSort(), threeWayQuickSortAlgo(), threeWayPartition() - functions used for three way quick sort algorithm

Table and Graph screenshots

Attached screenshots on the behavior of different sorting algorithms when the input size of the array changes.
We have taken array size of 1000, 5000, 10000

Table

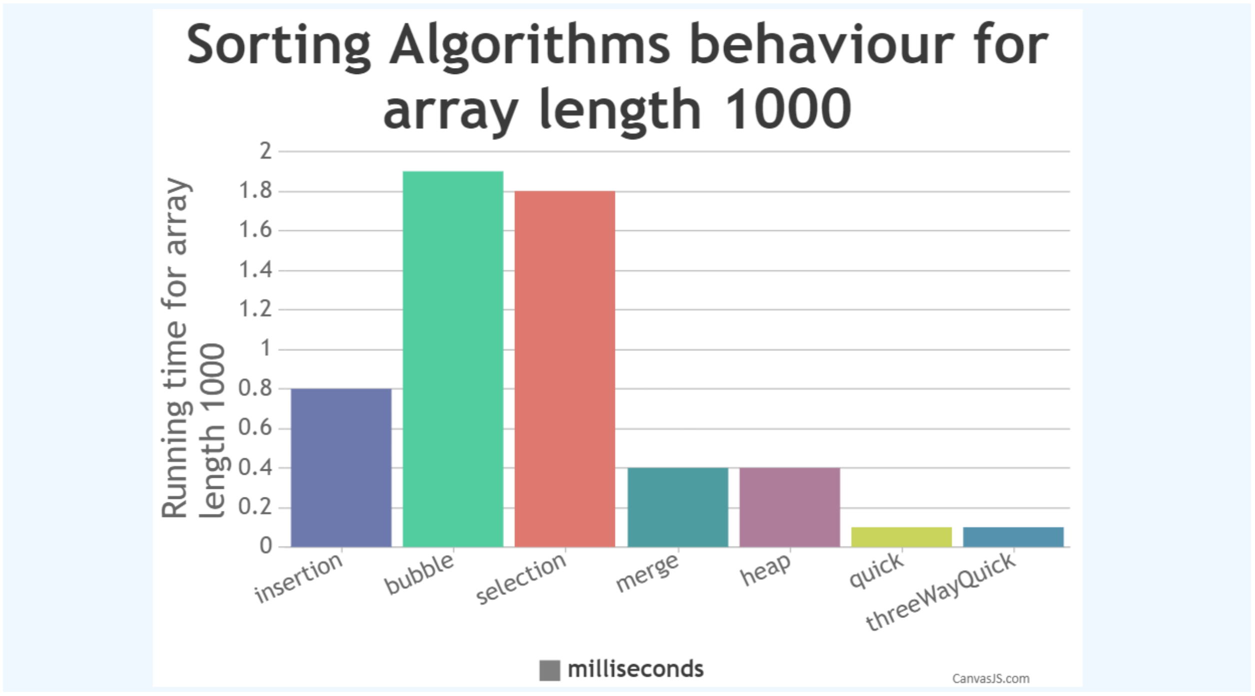


Running time comparision

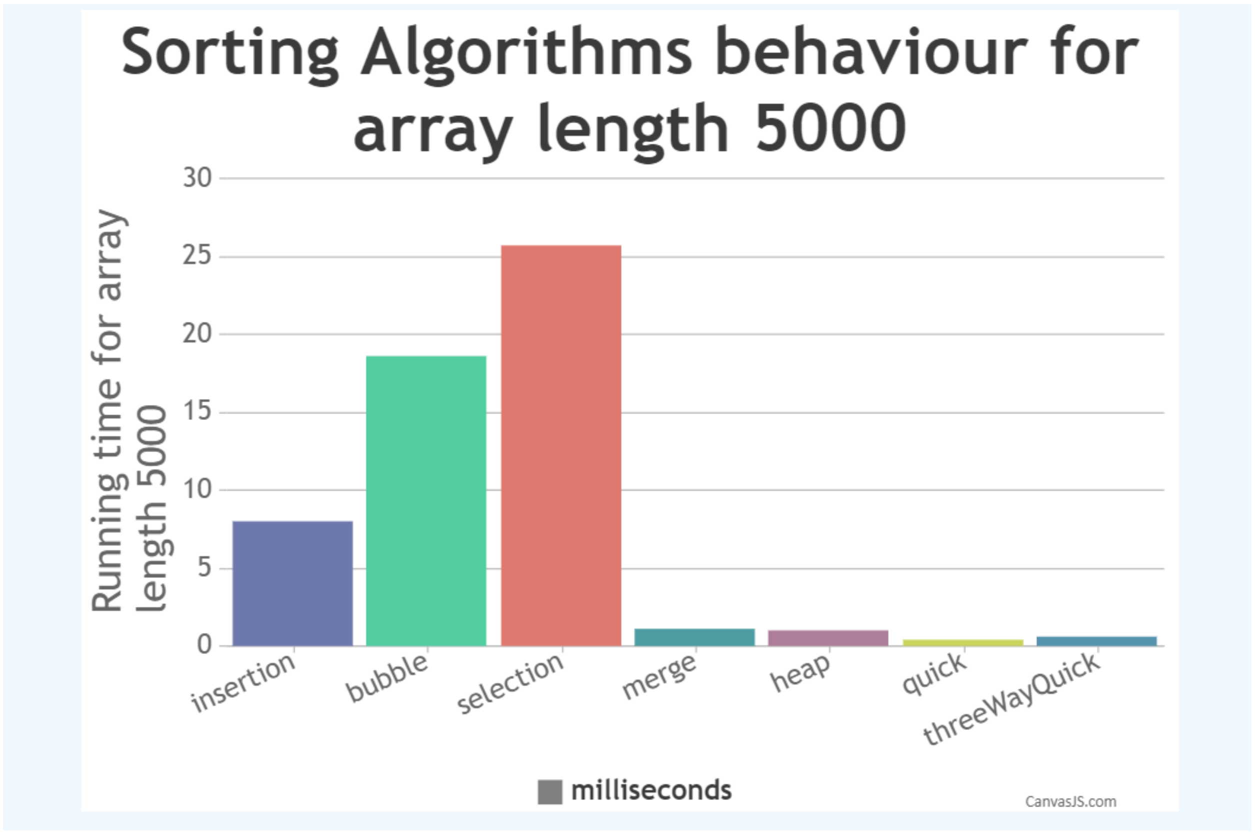
Input size	Insertion	Bubble	Selection	Merge	Heap	Quick	3 way Quick
1000	0.8000001907348633	1.8999996185302734	1.799999713897705	0.40000009536743164	0.40000009536743164	0.10000038146972656	0.09999990463256836
5000	8	18.59999990463257	25.699999809265137	1.0999999046325684	1	0.40000009536743164	0.5999999046325684
10000	48.59999990463257	118.5	173.40000009536743	3.9000000953674316	3	1.0999999046325684	1.3000001907348633

Graph

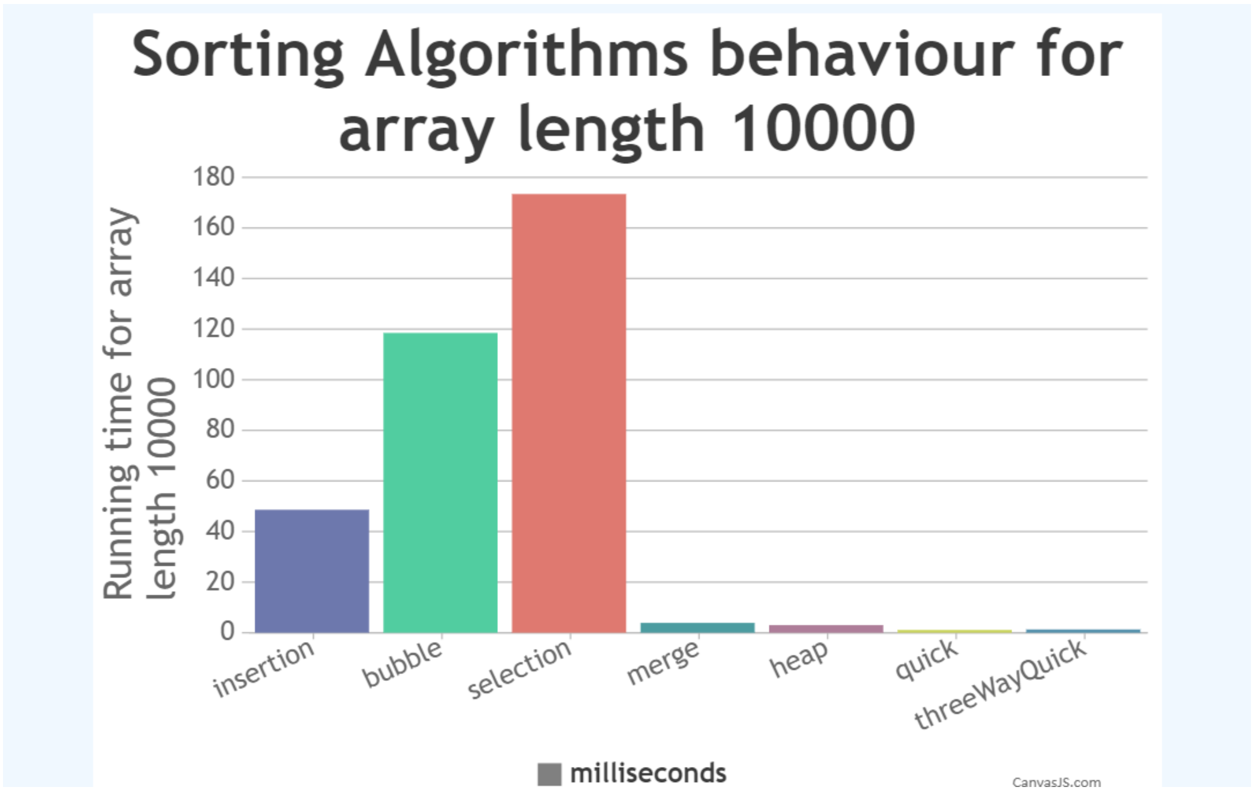
For array size - 1000



For array size - 5000



For array size - 10000



Conclusion
It is observed that as we increase the input array size, algorithms like three way quick sort, quick sort, heap sort, merge sort perform way better than algorithms like selection sort, bubble sort and insertion sort.