Computer & Systems Engineering Department

Data Structures and Algorithms

Implementing Binary Heap & Sorting Techniques

Contributors:

	Name	ID
1	Adel Mahmoud Mohamed Abdelrahman	20010769
2	Amr Ahmed Abdelazim	20011037
3	Marwan Essam Eldin	20011859
4	Mohamed Amr Abdelfattah	20011675
5	Mennat-Allah Mahmoud Saad	19016713

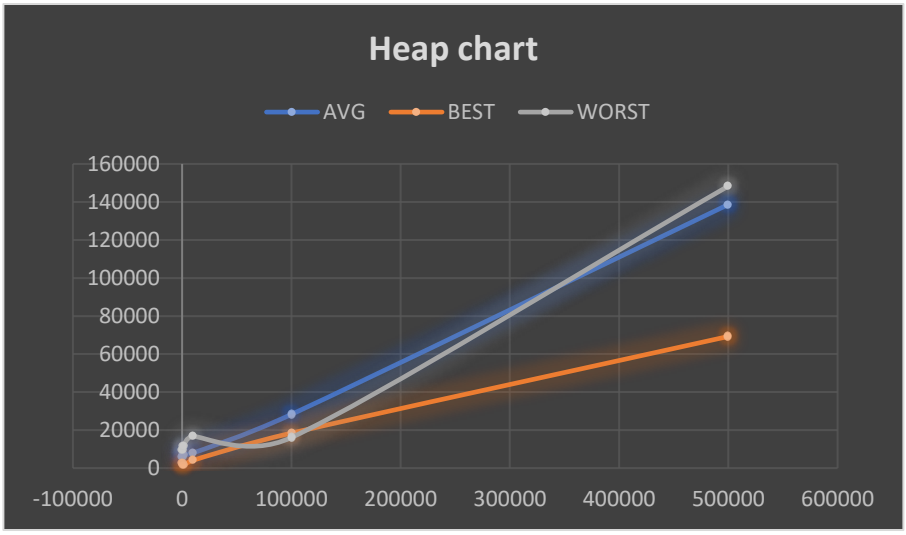
Time and Space Complexity of the Sorting algorithms:

Algorithm	*Space complexity*	*Time Complexity*
Counting sort	***O (N + K)***	***O(N + K)***
Merge sort	***O(N)***	***O (N Lg(N))***
Heap sort	***O(N)***	***O (N Lg (N))***
Bubble sort	***O (1)***	***O (N^2)***

Comparison between them according to the mean time to get the array sorted:

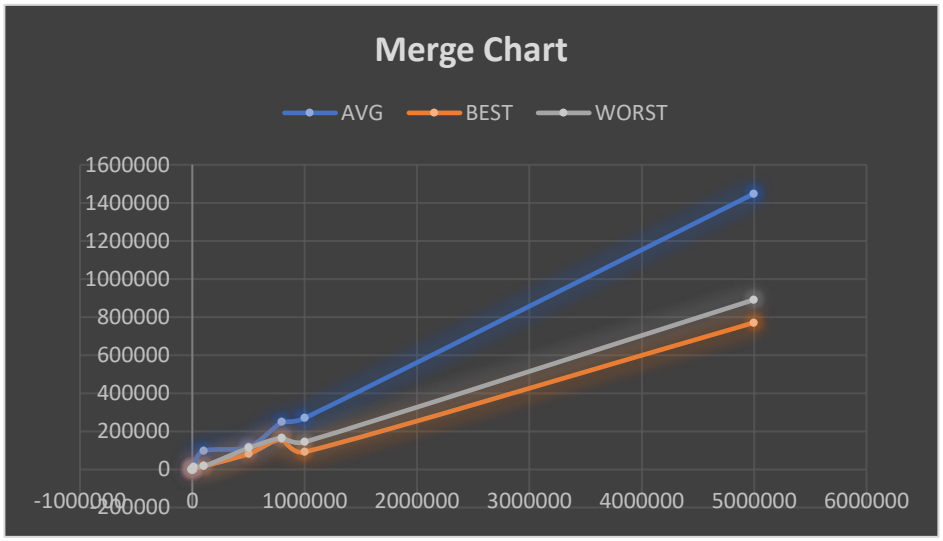
- **Heap sort:**

HEAP	AVG	BEST	WORST
100	6000.5	2731.7	9694.8
1000	6998.9	1465.9	11627.7
10000	7991.6	4143.6	16741
100000	28267.3	18434.8	16111
500000	138665.8	69260.7	148332



- **Merge sort:**

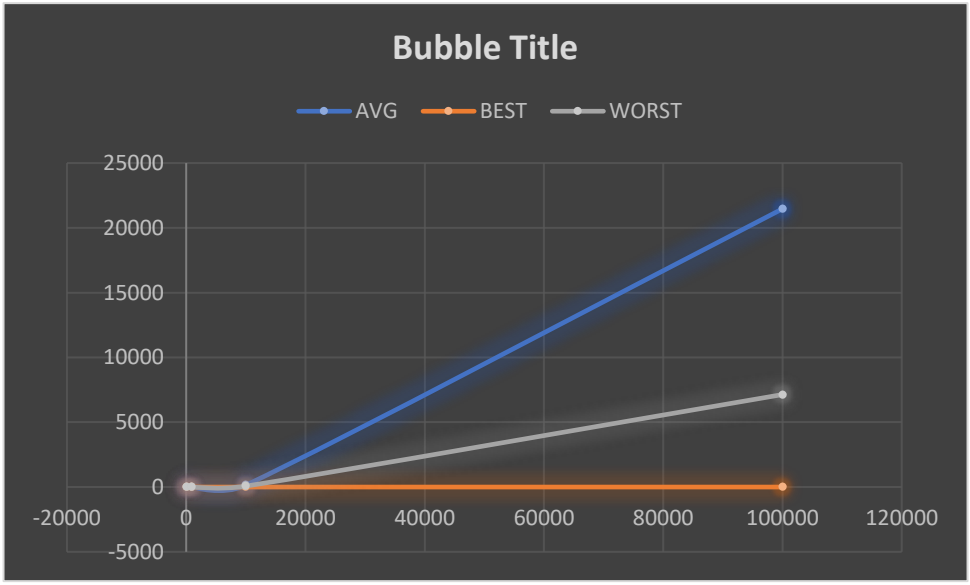
Merge	AVG	BEST	WORST
100	78.7	54.1	50.5
1000	350.7	298.8	691.2
10000	3748.2	1793.5	15452.2
100000	96379.2	16268	16356.9
500000	117842.4	81790.5	111814.9
800000	248391.2	160190.6	165734.7
1000000	269323.7	91707.1	144137.2
5000000	1448516	771022.4	891245.2
Mean	724297.5	385538.3	445647.9



- **Bubble sort:**

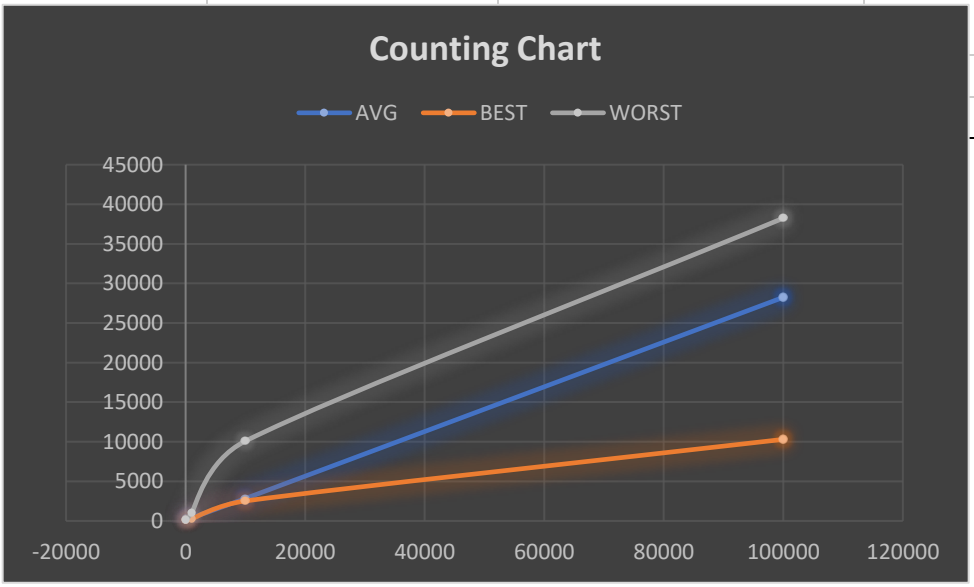
WITH Millis

Bubble	AVG	BEST	WORST
100	1	0	0
1000	5	0	1
10000	144	0	72
100000	21463.5	0	7136
Mean	10732.25	0	3568



- **Counting sort**

Counting	AVG	BEST	WORST
100	174.5	82.1	141.3
1000	405.1	243.3	1006.1
10000	2828.7	2520.9	10083.9
100000	28242.8	10299.8	38243.9
Mean	174.5	82.1	141.3



AVG	Heap	Counting	Merge	Bubble
100	6000.5	174.5	78.7	1 ms
1000	6998.9	405.1	350.7	5 ms
10000	7991.6	2828.7	3748.2	144 ms
100000	28267.3	28242.8	96379.2	21463.5 ms
500000	138665.8		117842.4	
800000	224709.3		248391.2	
1000000	261773.5		269323.7	
5000000	18411110		1448516.3	

BEST	Heap	Counting	Merge	Bubble(ms)
100	2731.7	82.1	54.1	0
1000	1465.9	243.3	298.8	0
10000	4143.6	2520.9	1793.5	0
100000	18434.8	10299.8	16268	0
500000	69260.7		81790.5	1
800000	138742.3		160190.6	
1000000	121719.6		91707.1	
5000000	591794.1		771022.4	

Worst	Heap	Counting	Merge	Bubble(Seconds)
100	9694.8	141.3	50.5	0ms
1000	11627.7	1006.1	691.2	1ms
10000	16741	10083.9	15452.2	72 ms
100000	16111	38243.9	16356.9	7136 ms
500000	148332		111814.9	
800000	134884.7		165734.7	
1000000	120964		144137.2	
5000000	927841.1		891245.2	

Conclusion:

In terms of time complexity and handling big sizes, Binary Heap and Efficient sort are more efficient than Non-Comparison Sort and Simple Sort. Binary Heap has a time complexity of O (log n) for insertion and deletion, while Efficient sort has a time complexity of O (n log n). Non-Comparison Sort algorithms like Counting Sort and Radix Sort have a linear time complexity of O(n+k), where k is the range of values in the input array. However, they require additional memory space to store the counts or buckets. Simple Sort algorithms like Bubble Sort and Selection Sort have a time complexity of O(n^2), which makes them inefficient for large input sizes. Therefore, Binary Heap and Efficient sort are preferred for handling large data sets with faster processing times.