Name: Hrishikesh Amulkumar Bodkhe

Enrolment No.: 2022CSM1006

Subject: CS506 – Data Structure (Lab Exercise 6)

System Specification: OS – Windows 11 64-bit, Processor - AMD Ryzen 5 5625U with Radeon Graphics 2.30 GHz, RAM – 8 GB.

Aim: To implement and analyze the time complexity of the Johnson algorithm using Array, Binary Heap, Binomial Heap, and Fibonacci Heap.

Programming Language: C++

Observations:

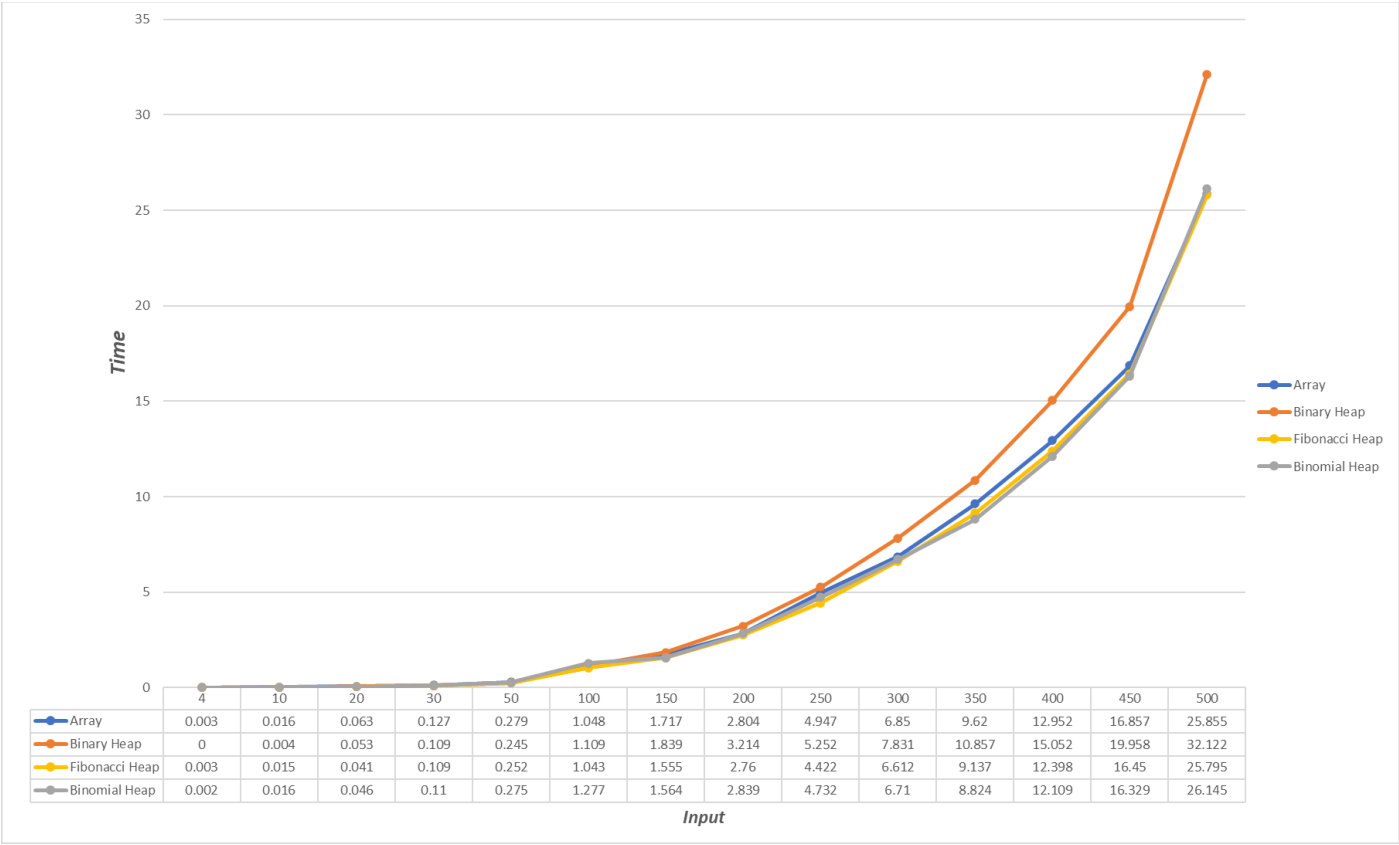
Details of the data:

Table 1 below (on page 2) shows the input sizes of the adjacency matrix and time in seconds for array, binary heap, binomial heap, and Fibonacci heap.

Taking into consideration the tabular data, a graph is constructed for the comparison of the running time of different data structures which is shown by **Graph 1 (**on page 3).

Table 1:-

Input Size	Array	Binary Heap	Binomial Heap	Fibonacci Heap
4	0.003	0	0.002	0.003
10	0.016	0.004	0.016	0.015
20	0.063	0.053	0.046	0.041
30	0.127	0.109	0.11	0.109
50	0.279	0.245	0.275	0.252
100	1.048	1.109	1.277	1.043
150	1.717	1.839	1.564	1.555
200	2.804	3.214	2.839	2.76
250	4.947	5.252	4.732	4.422
300	6.85	7.831	6.71	6.612
350	9.62	10.857	8.824	9.137
400	12.952	15.052	12.109	12.398
450	16.857	19.958	16.329	16.45
500	25.855	32.122	26.145	25.795



	4	10	20	30	50	100	150	200	250	300	350	400	450	500
Array	0.003	0.016	0.063	0.127	0.279	1.048	1.717	2.804	4.947	6.85	9.62	12.952	16.857	25.855
Binary Heap	0	0.004	0.053	0.109	0.245	1.109	1.839	3.214	5.252	7.831	10.857	15.052	19.958	32.122
Fibonacci Heap	0.003	0.015	0.041	0.109	0.252	1.043	1.555	2.76	4.422	6.612	9.137	12.398	16.45	25.795
Binomial Heap	0.002	0.016	0.046	0.11	0.275	1.277	1.564	2.839	4.732	6.71	8.824	12.109	16.329	26.145

Graph 1

As we can see from the above graph, for smaller inputs there is not much difference between the running times. But as the input increases, we can see that both binomial and fibonacci heaps perform well while binary heap is performing poor. Fibonacci heap performs well than others because the amortized time complexity of ***decrease_key*** function is O(1) while that of binomial and binary heaps is O(log N). Also finding the minimum element in heap is done in O(1) in Fibonacci heap but in binomial heap it is done in O(log N). Further Fibonacci heap is implemented as a circular linked list which allows more flexibility but binomial heap is implemented as a singly linked list.

Coming to binary heaps, it is performing poor as compared to binomial heaps because the ***meld*** or ***union*** operation takes O(1) time in binomial heaps while it takes O(log N) in Fibonacci heaps. Coming to the array vs binary heaps scene it is clearly seen that array implementation is taking

less time as compared to binary heaps because the implementation followed in the program is such that there is no decrease_key functionality used in the binary heap and thus lazy deletion is followed. That's why binary heaps are performing worst as compared to arrays.

Conclusion:-

Thus, after seeing the results it is concluded that Fibonacci and Binomial heaps perform better on large graphs.