CS201-HW5

Experimental Setup

In this homework, I compared algorithms which are AlgorithmSortAll, AlgorithmSortK, AlgorithmSortHeap and AlgorithmSortQuick to see which one is the fastest sort algorithm. The algorithms sort the inputs descending order and after sorting, they give the median element as the result.

First, I compared all of them with 50k, 100k, 150k, 200k and 250k inputs with 10 different random sets for each to avoid high variance and took the average time for each to compare.

After a quick look, to see which one is the fastest clearly, I compared AlgoritmSortHeap with AlgorithmSortQuick by massive input sizes which are 1M, 3M, 5M, 10M, 20M and 100M.

All of my inputs was chosen randomly the range between 0 to 1 million.

Result

My all tables are below

N = 50 000	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10	Average
AlgorithmSortAll	1.60	1.55	1.51	1.52	1.50	1.50	1.50	1.51	1.51	1.49	1.52
AlgorithmSortK	1.32	1.26	1.24	1.25	1.26	1.25	1.24	1.25	1.26	1.24	1.26
AlgorithmSortHeap	0.03	0.03	0.03	0.03	0.02	0.03	0.02	0.02	0.02	0.02	0.02
AlgorithmSortQuick	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02

N = 100 000	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10	Average
AlgorithmSortAll	6.36	6.01	5.97	6.01	5.96	6.01	5.98	5.99	5.97	5.96	6.02
AlgorithmSortK	5.15	5.00	4.96	5.01	4.94	4.96	4.94	4.95	4.96	4.96	4.98
AlgorithmSortHeap	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05
AlgorithmSortQuick	0.04	0.04	0.04	0.04	0.03	0.03	0.03	0.04	0.04	0.03	0.04

N = 150 000	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10	Average
AlgorithmSortAll	13.81	13.50	13.47	13.42	13.44	13.46	13.42	13.45	13.41	13.43	13.48
AlgorithmSortK	11.45	11.14	11.13	11.08	11.16	11.21	12.08	12.02	11.87	11.84	11.50
AlgorithmSortHeap	0.08	0.08	0.07	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07
AlgorithmSortQuick	0.06	0.06	0.05	0.05	0.06	0.06	0.05	0.05	0.05	0.05	0.06

N = 200 000	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10	Average
AlgorithmSortAll	24.76	23.82	23.97	22.40	22.38	22.30	22.28	22.32	22.22	22.65	22.91
AlgorithmSortK	19.99	19.68	19.97	19.65	19.71	19.65	19.85	19.58	19.62	19.69	19.74
AlgorithmSortHeap	0.13	0.11	0.10	0.09	0.09	0.09	0.09	0.10	0.10	0.10	0.10
AlgorithmSortQuick	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07

N = 250 000	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10	Average
AlgorithmSortAll	37.50	37.37	39.72	41.30	39.13	39.07	48.35	45.05	60.61	66.89	45.50
AlgorithmSortK	31.41	31.00	30.82	30.92	31.61	31.60	30.80	31.40	33.36	32.80	31.57
AlgorithmSortHeap	0.13	0.12	0.14	0.12	0.13	0.12	0.14	0.14	0.15	0.12	0.13
AlgorithmSortHeap	0.10	0.09	0.09	0.10	0.10	0.09	0.09	0.08	0.09	0.09	0.09

Table 1 – All randomly chosen cases for each input size

N	K	AlgorithmSortAll	AlgorithmSortK	AlgorithmSortHeap	AlgorithmSortQuick
50,000	25,000	1.518499 s	1.256806 s	0.0247994 s	0.0185383 s
100,000	50,000	6.022519 s	4.984001 s	0.0514799 s	0.0369444 s
150,000	75,000	13.48129 s	11.49845 s	0.0722134 s	0.055359 s
200,000	100,000	22.91054 s	19.73893 s	0.1008832 s	0.0709403 s
250,000	125,000	45.49803 s	31.57303 s	0.1320137 s	0.0909974 s

Table 2 – Average time for each input size

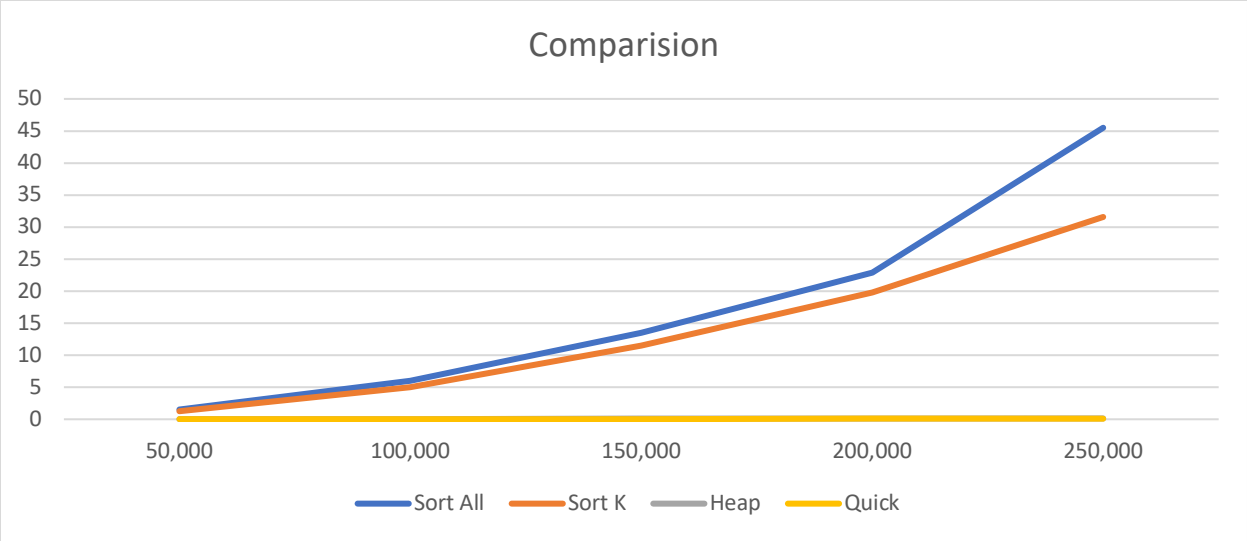
N	K	AlgorithmSortHeap	AlgorithmSortQuick
1,000,000	500,000	0.553643 s	0.396725 s
3,000,000	1,500,000	1.70424 s	1.09269 s
5,000,000	2,500,000	2.82194 s	1.83729 s
10,000,000	5,000,000	5.85884 s	3.6345 s
20,000,000	10,000,000	11.4468 s	7.06025 s
100,000,000	50,000,000	64.1725 s	33.4557 s

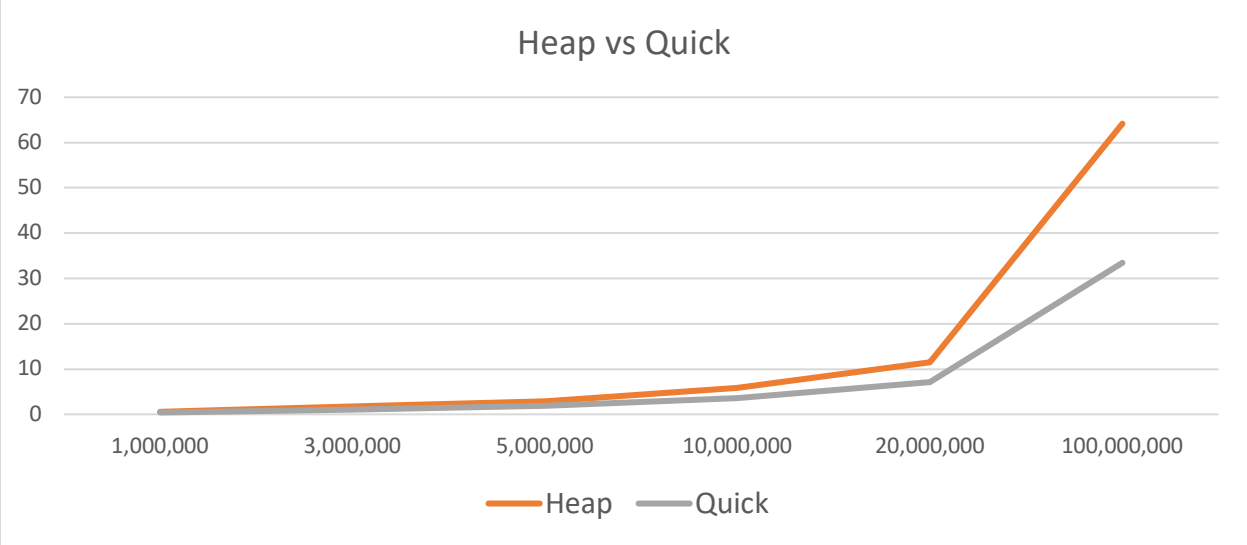
Table 3 – Heap vs Quick with huge input sizes

To generalize, Quick sort performs best performance for each input sizes, and then Heap sort comes and then sort K and sort All come in order.

$$T_{\text{Quick Sort}} < T_{\text{Heap Sort}} < T_{\text{Sort K}} < T_{\text{Sort All}}$$

(Note: T denotes time)





Discussion

Theoretically, Sort All and Sort K possess same time complexity O(n^2) but Sort K is little bit faster because it does not store all the numbers unlike Sort All. In first chart Sort All and Sort K are like O(n) but after 200k they act like O(n^2). In the second chart, theoretically Heap and Quick have O(NlogN) time complexity but our implementation makes Quick sort O(N). Therefore, Quick sort is clearly fastest algorithm compared to other 3 algorithms.

Mustafa Çamlı – S014665