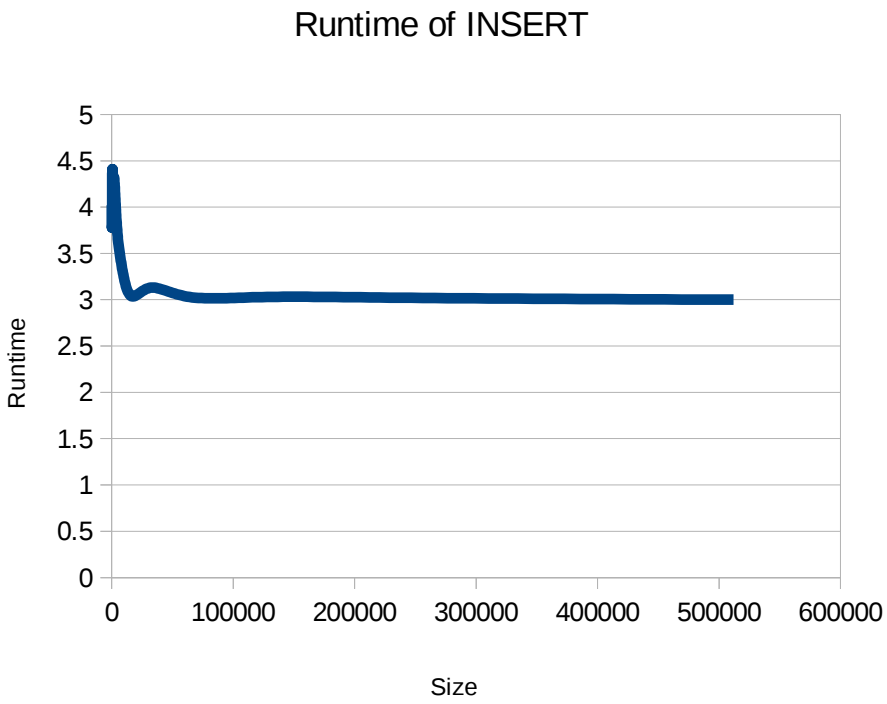
CS583 Project Report – Violation Heap

Haoliang Wang (G00840008)

In this project, the Violation Heap proposed by Amr Elmasry is implemented and its runtime is empirically verified. Three operations – INSERT, DECREASE-KEY, EXTRACT-MIN are tested with Size varying from 100 to 512000 and results are shown as follows.

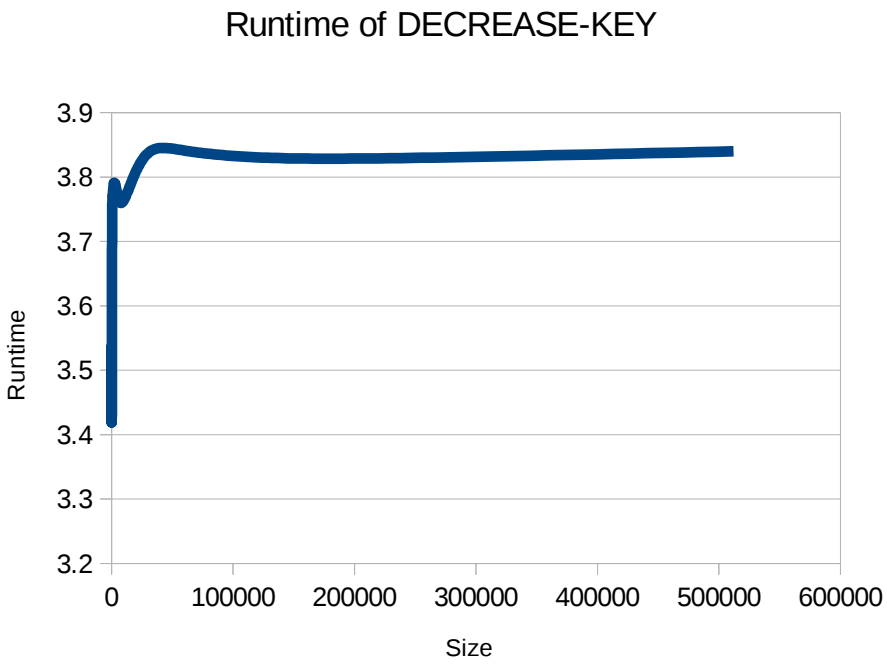
1. Runtime of INSERT

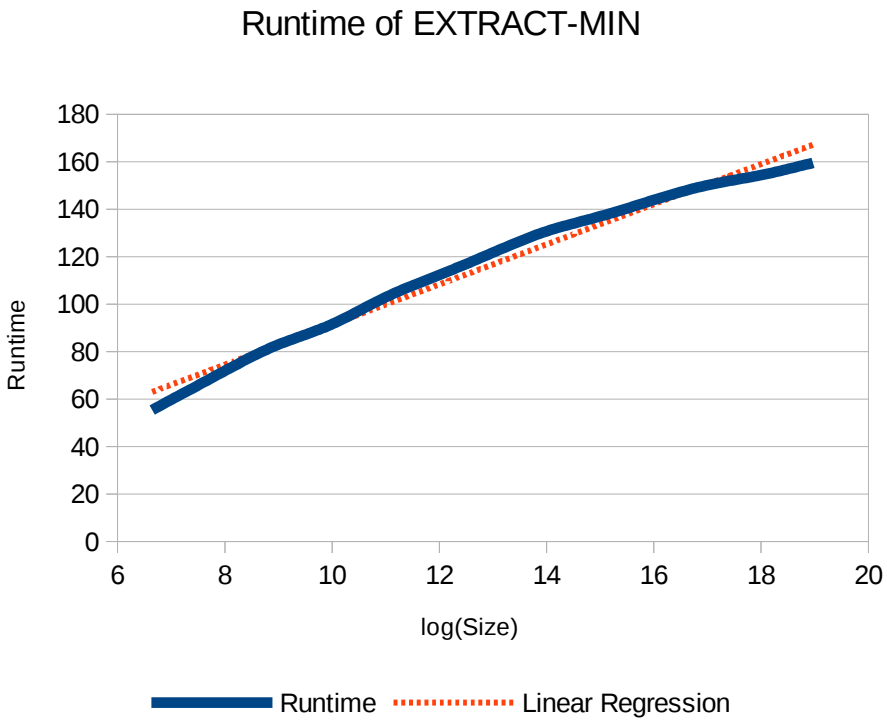


As we can see from the figure above, except for some noises when size is small (below 10000 due to the randomness of choosing which key to decrease), the runtime of INSERT remains to be a constant (around 3) with the increase of problem size, which indicates its amortized runtime is O(1).

2. Runtime of DECREASE-KEY

As we can see from the figure below, similar to the runtime of INSERT, the runtime of DECREASE-KEY remains to be constant despite the increase of size, indicating its amortized runtime is also O(1).





3. Runtime of EXTRACT-MIN

As shown in the figure above, with the x axis to be logarithmic, the runtime of EXTRACT-MIN is almost linear to log(Size) (Compared with the linear regression line), from which we can conclude that the runtime of EXTRACT-MIN is O(log n).

The complete experiment data (problem size and the corresponding amortized runtime) is shown in the table below.

SIZE	100	250	500	1000	2000	4000	8000
INSERT	4.02	3.81	4.38	4.28	4.31	3.83	3.38
DECREASE_KEY	3.54	3.43	3.70	3.77	3.79	3.78	3.76
EXTRACT_MIN	55.38	71.47	82.66	91.24	102.58	111.99	121.39

SIZE	16000	32000	64000	128000	256000	512000
INSERT	3.04	3.13	3.03	3.03	3.02	3.00
DECREASE_KEY	3.79	3.84	3.84	3.83	3.83	3.84
EXTRACT_MIN	130.42	136.78	143.63	149.93	154.22	159.53

4. Conclusion

From the experiment result shown above, performance three operations in the Violation Heap (INSERT, EXTRACT-MIN and DECREASE-KEY) is verified to have the same performance guarantees as Fibonacci heaps.