

EECS 560 - Lab #9

Theodore Lindsey

October 17, 2015

1 Overall Organization of Experiment

For this lab, we attempt to evaluate whether a skew heap or a leftist heap performs more quickly provided the same records are inserted into both tables. We also attempt to evaluate which type of heap performs random inserts and deletemins more efficiently.

2 Data Generation

In this experiment, it is necessary that we insert the same values into both the leftist heap and the skew heap in order to achieve an apple-to-apples comparison. To this end, we will use identical seeds for our pseudo random number generator in order to ensure that the same sequence of “random” numbers are inserted into both heaps.

In order to reduce variation in the results, multiple trials are run for each quantity of records. Each trial uses a different seed in order to reduce the impact of troublesome values to be inserted that might occur from a particular seed. Each of the seeds used is the same for corresponding trials in each record quantity group.

Additionally, following a uniform distribution, the testing function randomly either inserts random records into the heap or deletes the minimum value from the heap. The insertions are interspersed with the deletemin operations in order to maintain, roughly, the size of the heap.

Timing is measured by the provided timer class. Times for individual runs are recorded in the log file ('log.csv'). Averages for a specific quantities of records and merge method are calculated after running each experiment. The averages are then displayed to console. Averages aren't recorded in the log file in order to make it easier and cleaner to import into a database or spreadsheet.

3 Summary of Results (CPU Timing)

In all trials, the skew heap had better performance than the corresponding leftist heap (as seen in each of the figures).

4 Observation and Conclusion

As noted in the previous section, the skew heap had better performance in both the build and insert/deletemin trials. This is probably because it takes more work to compute and then compare rank in order to decide if a swap is needed than it is to just swap every time. This is also consistent with what we learned from lecture.

5 Data

Seed	n	Leftist	Skew
0	50000	0.01652	0.01125
1	50000	0.01521	0.01068
2	50000	0.01477	0.01049
3	50000	0.01558	0.01074
4	50000	0.01452	0.01036
5	50000	0.01578	0.01056
avg		0.01540	0.01068
0	100000	0.03346	0.02436
1	100000	0.03139	0.02399
2	100000	0.03229	0.02510
3	100000	0.03160	0.02495
4	100000	0.03148	0.02391
5	100000	0.03156	0.02417
avg		0.03196	0.02441
0	200000	0.06861	0.05192
1	200000	0.06434	0.05501
2	200000	0.06724	0.05810
3	200000	0.06598	0.05643
4	200000	0.06642	0.05654
5	200000	0.06807	0.05904
avg		0.06678	0.05617
0	400000	0.15191	0.11320
1	400000	0.12701	0.11636
2	400000	0.13138	0.11915
3	400000	0.13366	0.11474
4	400000	0.13650	0.10079
5	400000	0.12776	0.11127
avg		0.13470	0.11258

Figure 1: Build times with 5 seed trials per value of n .

Seed	n	Leftist	Skew
0	50000	0.00369	0.00197
1	50000	0.00374	0.00197
2	50000	0.00378	0.00197
3	50000	0.00375	0.00196
4	50000	0.00376	0.00195
5	50000	0.00377	0.00195
avg		0.00375	0.00196
0	100000	0.00819	0.00450
1	100000	0.00828	0.00451
2	100000	0.00833	0.00453
3	100000	0.00835	0.00450
4	100000	0.00840	0.00451
5	100000	0.00838	0.00452
avg		0.00832	0.00451
0	200000	0.01808	0.01014
1	200000	0.01826	0.01005
2	200000	0.01836	0.01006
3	200000	0.01978	0.01007
4	200000	0.01853	0.01006
5	200000	0.01851	0.01003
avg		0.01859	0.01007
0	400000	0.04229	0.02038
1	400000	0.03797	0.02006
2	400000	0.03838	0.02002
3	400000	0.03862	0.02001
4	400000	0.06750	0.02015
5	400000	0.03837	0.02000
avg		0.04385	0.02010

Figure 2: Insertion and deletemin times with 5 seed trials per value of n .

6 Visualisation of Data

Figure 3 shows the average completion times for building the heaps for each seed for a given number of records. Figure 4 shows the average completion times for each seed for performing insertions and deletemin operations for a given number of records. As seen in the figures below, the results of this experiment are consistent with what we expected from the discussion in lecture.

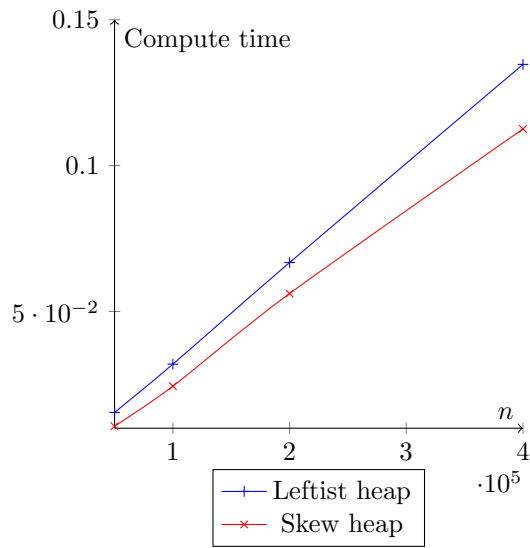


Figure 3: Building times.

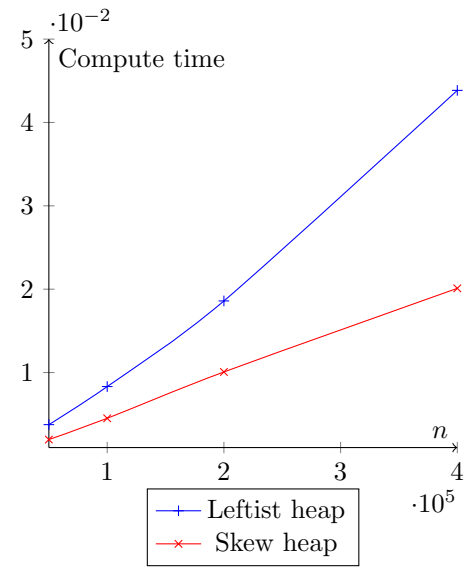


Figure 4: Insert / deletemin times.