Overall Organization

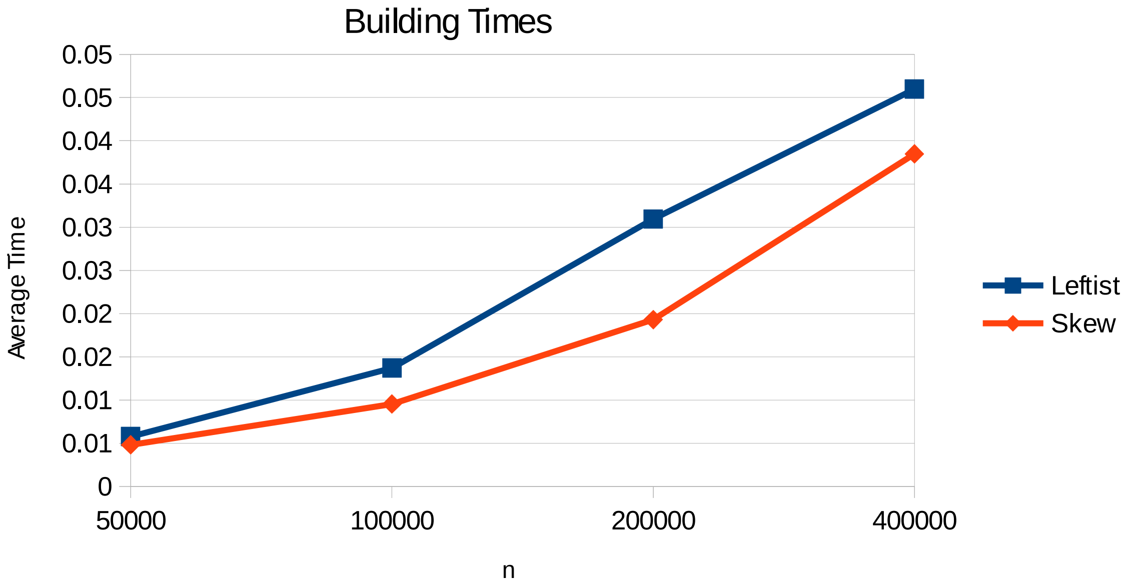
I structured my experiment by using nested for loops. My outermost loop ran 5 times in order to accommodate the 5 trials (using 5 different seeds), The 5 different seeds that were used were simply the loop iteration (i.e. on the first loop it would use a seed of 1 etc…). Then the next loop would go through the values of n, starting from 500000 going to 400000. The innermost loop would run the insert or whatever function I was testing. First runs did building, the second run randomly generated a choice between doing an insert and a deletemin, starting with the Leftist Tree then running the trials for the Skew Heap.

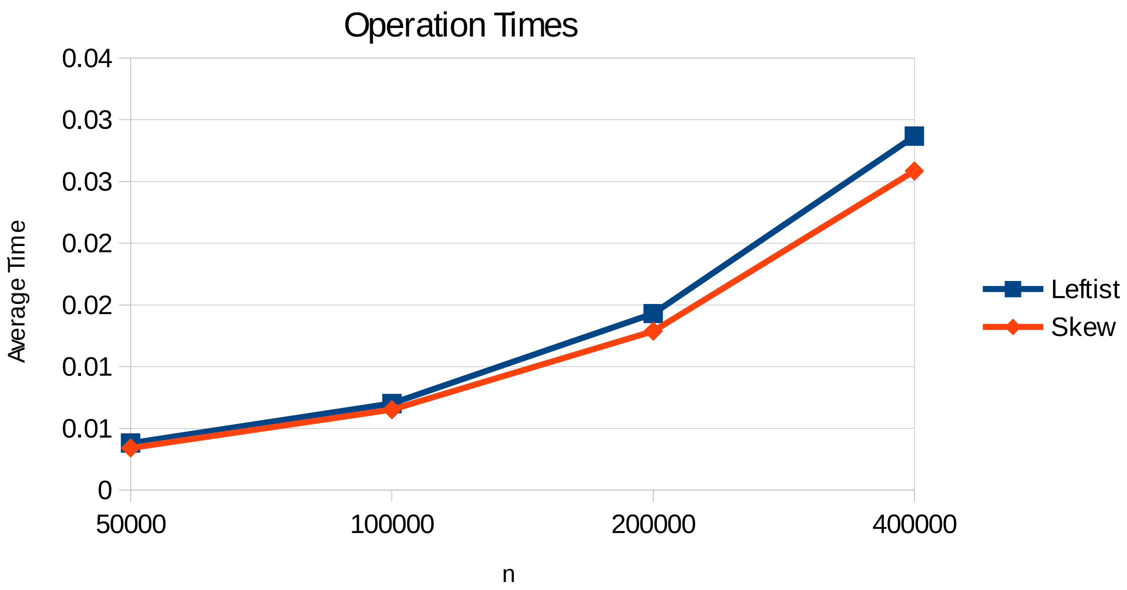
Raw Data

Leftist Tree Building with n = 50000 **Avg Time: 0.005782**
0.005639
0.006453
0.005633
0.009476
0.005603
Leftist Tree Building with n = 100000 **Avg Time: 0.013712**
0.014418
0.014256
0.011198
0.014714
0.011194
Leftist Tree Building with n = 200000 **Avg Time: 0.030944**
0.025275
0.032291
0.031149
0.032116
0.031298
Leftist Tree Building with n = 400000 **Avg Time: 0.045980**
0.04572
0.046432
0.045746
0.04633
0.045707
Skew Heap Building with n = 50000 **Avg Time: 0.004823**
0.004619
0.004608
0.005227
0.005177
0.00464
Skew Heap Building with n = 100000 **Avg Time: 0.009543**
0.010373
0.009217
0.009231
0.010457

0.009197
Skew Heap Building with n = 200000 **Avg Time: 0.019312**
0.019241
0.01901
0.01973
0.019277
0.018963
Skew Heap Building with n = 400000 **Avg Time: 0.038478**
0.038115
0.038217
0.038802
0.038858
0.038256
Leftist Heap with n = 50000 **Avg Time: 0.003812**
0.004192
0.003483
0.003987
0.00344
0.003457
Leftist Heap with n = 100000 **Avg Time: 0.007012**
0.00693
0.006886
0.007428
0.007318
0.006968
Leftist Heap with n = 200000 **Avg Time: 0.014299**
0.014312
0.014258
0.014419
0.014274
0.013937
Leftist Heap with n = 400000 **Avg Time: 0.028664**
0.028481
0.028775
0.028651
0.028943
0.028698
Skew Heap Operations with n = 50000 **Avg Time: 0.003409**
0.003142
0.00313
0.003125
0.004263
0.003118
Skew Heap Operations with n = 100000 **Avg Time: 0.006523**
0.006638
0.006191
0.006229
0.006815
0.006655

Skew Heap Operations with n = 200000 **Avg Time: 0.012875**
0.012431
0.013031
0.012866
0.012535
0.013418
Skew Heap Operations with n = 400000 **Avg Time: 0.025846**
0.025499
0.025976
0.025433
0.025912
0.025995





Summary of Results

The Skew Heap seemed to outperform the Leftist Tree in both building and operation times, averaging a lower time for every value of n for both. The difference was more apparent for larger values of n, as for the lower values of n, the times were still rather close.

Observation and conclusion

It seems that the Skew Heap is simply a more efficient version of the leftist heap. By indiscriminately swapping the merging is a lot faster while still being able to maintain the position of the minimum value and also has the plus of not having to keep track of the rank.