Brent Mitchell
2564353

This experiment was to first make a binomial queue. The operations included insert, deletemin, and levelorder traversal for it. After creating the binomial queue, testing was needed to compare the timing of the minimum leftist heap, minimum skew heap, and minimum binomial queue. The tests were done using pseudo random numbers using the srand and rand C++ functions. Random numbers were first inserted into the leftist heap, skew heap, and binomial queue. Then a combination of insertions and deletions were performed on the three structures based on a probability function that was calculated from the pseudo random numbers.

Data was generated in three main loops. The outer loop generated the number of insertions into the structures, but only 10% of this total number was used for the insertion and deletion inner loop. The number of insertions started at 50,000 and doubled until 400,000 items were inserted into the structure. The second inner loop generated a seed to be used for the pseudo random number. This experiment used seed values of 1 to 5. The four inner loops were for the insertion only loop and the combination of insertion and deletion loop (two loops for each structure). It printed out the execution time of each inner loop. These times were put into an Excel spreadsheet to generate the graphs below comparing the execution time vs. number of insertions.

The binomial queue performed better in the insertion test, but the minimum leftist heap performed better in the insertion and deletion test. The execution time for all structures took longer as n increased. The three structures demonstrated O(log n) performance even though the worst case skew heap performance is O(n). The skew heap can have a worst case scenario of O(n) if it is a skew tree. If there will be a large amount of insertions and few deletions, then the binomial queue would be the better choice. The leftist heap would be better if there are going to be a lot of insertions and deletions. The average execution times are in the table below as well as the graphs comparing the two structures.

n	MinLeftistHeap Insert (s)	MinLeftistHeap Insert and Delete (s)	SkewHeap Insert (s)	SkewHeap Insert and Delete (s)	Binomial Queue Insert (s)	Binomial Queue Insert and Delete (s)
50000	0.014102	0.0006596	0.0151198	0.000595	0.0059002	0.0860242
100000	0.0302328	0.0013236	0.0341926	0.0012214	0.0136608	0.1708706
200000	0.0638686	0.002344	0.0722374	0.0022348	0.0347988	0.3097442
400000	0.1371172	0.0048716	0.163993	0.0045158	0.0781248	0.6227656

Brent Mitchell
2564353