

EECS 560: Lab 09 Report

Complexity Analysis of a LeftistHeap VS. SkewHeap

Aubrey Bud Linville

2819130

1. Overall Organization

This experiment was written in C++, with the actual complexity analysis portion of the experiment written in .cpp files that were instantiated by the project's makefile. This project mimics the behavior of a scheduler in an operating system and required two abstract data types – a LeftistHeap and a SkewHeap. Both of these heaps were implemented with a linked-node implementation. To implement the node, I used a struct with four instance variables – leftChild, rightChild, rank, and task. The rank variable was only needed for the LeftistHeap implantation, as SkewHeaps use a different ordering mechanism. I also had to implement an application, scheduler, and task class that further aids in the simulation of the operation system's scheduler. Data is read in from a text file specified by the user from a command line parameter. If no input file is given, my program uses default a *data.txt* file included in the package. The data is stored in an array of tasks that is then passed to the heaps' respective constructors so that it can build the proper heap. All typical heap functionality is implemented including deletion, insertion, and concatenating methods. The analysis files calculate average times that their respective functions take and displays it to the terminal.

2. Tabulated Data

Leftist Heap Complexity Analysis Timings									
addElem()									
Size:	10	50	100	500	1000	5000	10000	50000	100000
Test	Times (ms)								
T1	1	1	2	10	7	40	87	963	2077
T2	1	1	1	10	6	41	86	948	1991
T3	1	1	2	10	6	39	86	542	1489
T4	1	1	1	9	5	39	84	465	1173
T5	1	1	3	10	6	37	84	458	1154
T6	1	2	2	9	7	38	84	458	1115
T7	1	1	2	11	5	38	84	458	1123
T8	1	2	2	10	6	38	84	459	1352
T9	1	1	2	10	7	37	85	456	1696
T10	1	1	2	10	5	38	84	455	1696
T(AVG)	1.0	1.2	1.9	9.9	6.0	38.5	84.8	566.2	1486.6
deleteMinElem()									
Size:	10	50	100	500	1000	5000	10000	50000	100000
Test	Times (ms)								
T1	4	3	4	4	6	37	81	457	1050
T2	0	2	2	3	7	36	81	452	1069
T3	1	2	3	4	6	37	82	452	1101
T4	0	1	3	4	6	36	81	451	1020
T5	0	2	4	4	5	36	81	455	1093
T6	1	2	3	3	6	37	81	454	1092
T7	0	1	3	4	6	36	82	447	1011
T8	1	1	3	4	5	36	81	452	1063
T9	0	2	1	4	5	36	81	450	1123
T10	1	1	1	4	4	36	80	451	1020
T(AVG)	0.8	1.7	2.7	3.8	5.6	36.3	81.1	452.1	1064.2
concat()									
Size:	10	50	100	500	1000	5000	10000	50000	100000
Test	Times (ms)								
T1	0	1	2	7	26	90	421	2125	4296
T2	0	0	1	7	15	202	430	2151	4201
T3	0	1	1	7	16	132	423	2117	4331
T4	0	0	1	7	14	133	423	2133	4271
T5	0	1	1	7	16	194	421	2115	4283
T6	0	1	1	8	18	166	427	2127	4198
T7	0	1	1	7	14	144	420	2123	5334
T8	0	1	2	7	34	99	412	2129	4378
T9	0	1	1	7	15	158	409	2118	4395
T10	0	0	1	7	15	219	417	2102	4178
T(AVG)	0.0	0.7	1.2	7.1	18.3	153.7	420.3	2124.0	4386.5

Skew Heap Complexity Analysis Timings

addElem()

Size:	10	50	100	500	1000	5000	10000	50000	100000
Test	Times (ms)								
T1	1	0	1	5	10	52	238	1113	2477
T2	0	0	1	4	9	51	186	1160	2418
T3	1	0	0	4	10	50	121	752	2506
T4	0	1	1	4	9	50	127	712	2482
T5	0	0	0	4	9	50	118	653	2471
T6	0	1	1	4	8	49	118	639	2510
T7	1	0	1	4	9	50	105	963	2506
T8	0	0	0	4	9	49	120	979	2445
T9	0	0	1	4	9	50	119	920	2384
T10	1	1	1	6	9	50	120	886	2333
T(AVG)	0.4	0.3	0.7	4.3	9.1	50.1	137.2	877.7	2453.2

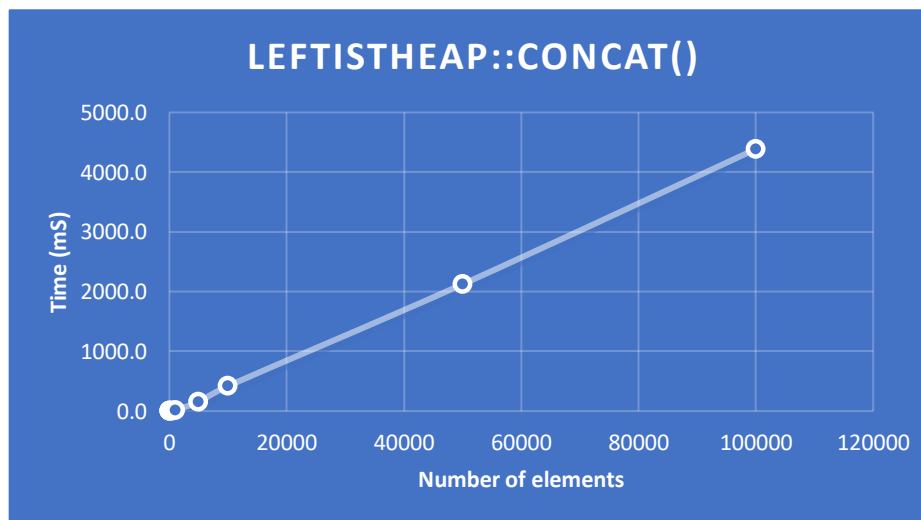
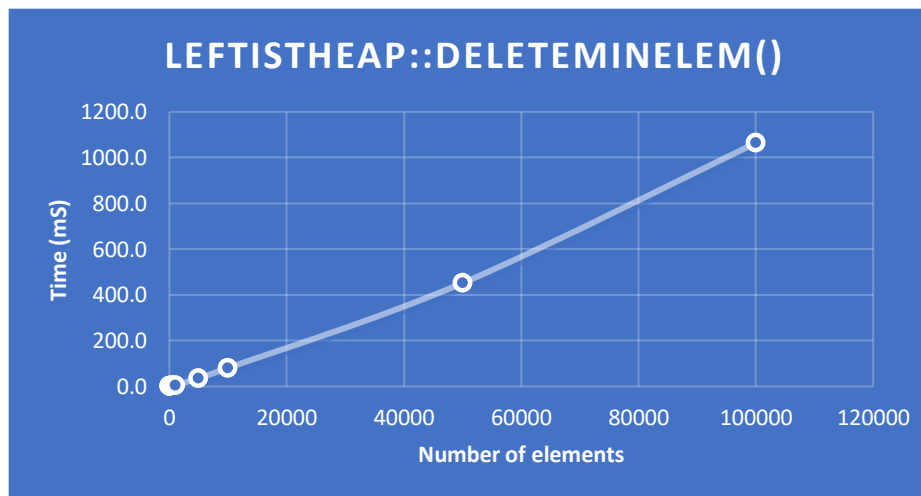
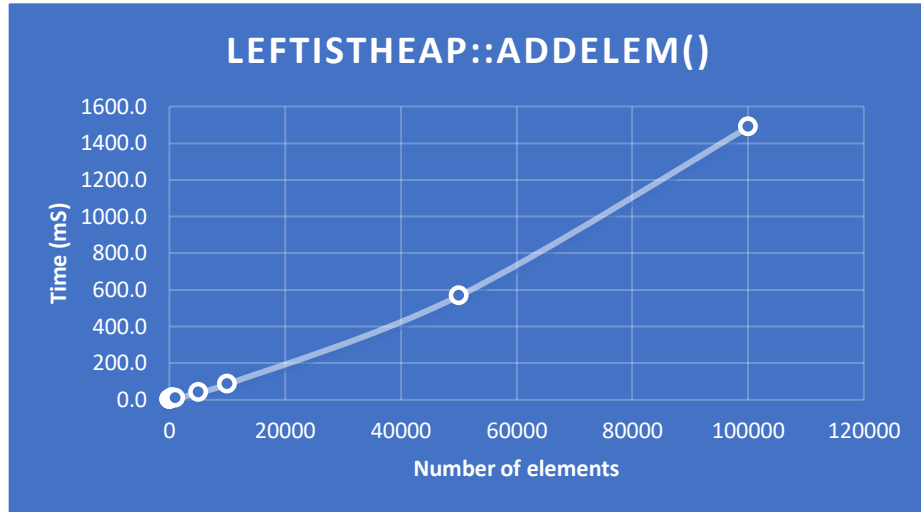
deleteMinElem()

Size:	10	50	100	500	1000	5000	10000	50000	100000
Test	Times (ms)								
T1	0	1	1	7	11	134	245	1106	2155
T2	1	0	1	6	15	134	242	1121	2159
T3	0	1	1	7	13	134	244	1105	2165
T4	0	0	1	6	16	131	245	1100	2155
T5	0	1	1	7	16	133	246	1103	2162
T6	0	1	2	8	12	230	240	1110	2173
T7	0	0	1	8	11	135	242	1115	2158
T8	0	0	1	6	13	137	238	1108	2171
T9	0	0	1	6	11	134	244	1113	2164
T10	1	0	1	7	11	147	247	1100	2152
T(AVG)	0.2	0.4	1.1	6.8	12.9	144.9	243.3	1108.1	2161.4

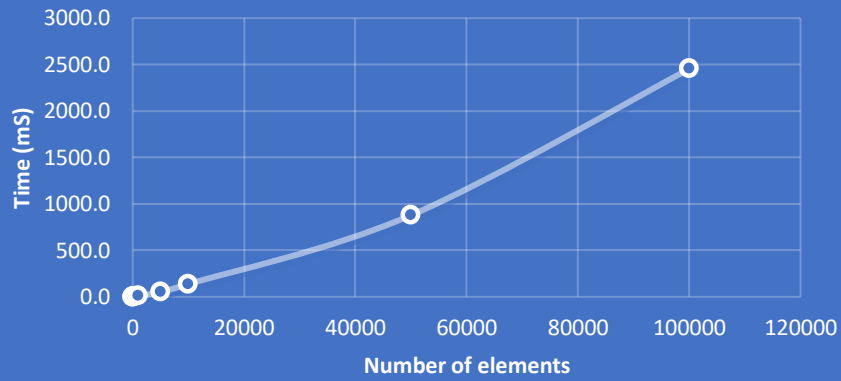
concat()

Size:	10	50	100	500	1000	5000	10000	50000	100000
Test	Times (ms)								
T1	0	1	1	6	16	110	331	2153	4397
T2	1	0	2	6	14	166	365	2181	4501
T3	0	0	1	7	14	112	360	2136	4302
T4	0	1	1	7	15	110	348	2131	4311
T5	0	0	1	6	13	144	338	2144	4463
T6	0	1	1	7	15	140	350	2150	4382
T7	0	0	2	7	13	115	324	2193	4378
T8	0	1	1	7	20	111	329	2143	4427
T9	0	0	1	7	14	114	329	2136	4356
T10	1	1	1	7	14	148	333	2140	4443
T(AVG)	0.2	0.5	1.2	6.7	14.8	127.0	340.7	2150.7	4396.0

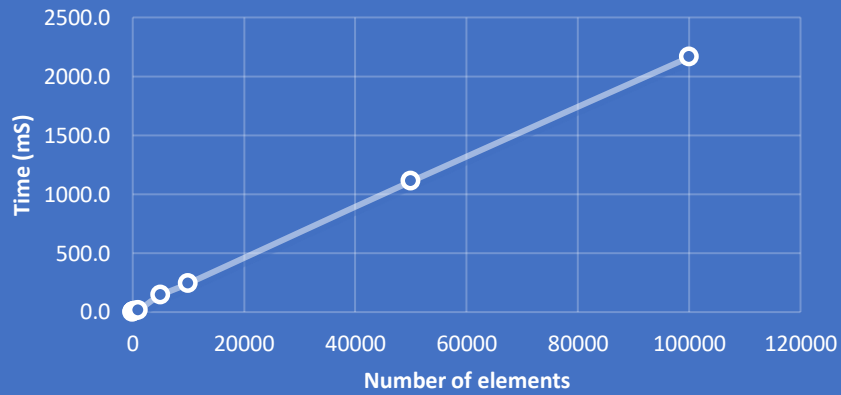
3. Graphs



SKEWHEAP::ADDELEM()



SKEWHEAP::DELETEMINELEM()



SKEWHEAP::CONCAT()

