

Austin Bailey  
EECS 560  
10/17/16

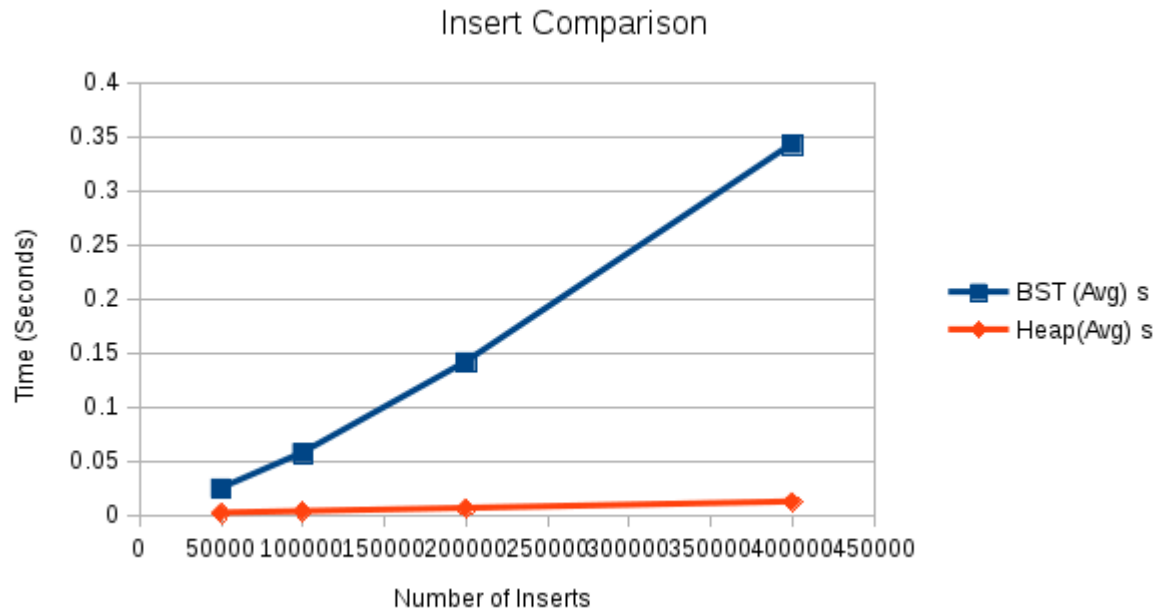
### Lab 6 Report

Note: The following Table was generated from a single test of a 581 Lab computer, rerun results may vary based on processing capability

Heap or BST	Operation	N	Trial Number	Time (sec)
BST	Insert	50000	1	0.02357
BST	Insert	50000	2	0.024467
BST	Insert	50000	3	0.024811
BST	Insert	50000	4	0.025345
BST	Insert	50000	5	0.022999
BST	Insert	50000	Average	0.0242384
BST	Insert	100000	1	0.05833
BST	Insert	100000	2	0.055432
BST	Insert	100000	3	0.058579
BST	Insert	100000	4	0.056718
BST	Insert	100000	5	0.05535
BST	Insert	100000	Average	0.0568818
BST	Insert	200000	1	0.136883
BST	Insert	200000	2	0.141891
BST	Insert	200000	3	0.142034
BST	Insert	200000	4	0.141209
BST	Insert	200000	5	0.142709
BST	Insert	200000	Average	0.140945
BST	Insert	400000	1	0.33288
BST	Insert	400000	2	0.340939
BST	Insert	400000	3	0.343733
BST	Insert	400000	4	0.341185
BST	Insert	400000	5	0.353148
BST	Insert	400000	Average	0.342377
Heap	Insert	50000	1	0.001598
Heap	Insert	50000	2	0.00159
Heap	Insert	50000	3	0.001582

Heap	Insert	50000	4	0.001588
Heap	Insert	50000	5	0.001651
Heap	Insert	50000	Average	0.0016018
Heap	Insert	100000	1	0.003054
Heap	Insert	100000	2	0.00296
Heap	Insert	100000	3	0.002939
Heap	Insert	100000	4	0.003042
Heap	Insert	100000	5	0.002964
Heap	Insert	100000	Average	0.0029918
Heap	Insert	200000	1	0.005901
Heap	Insert	200000	2	0.005902
Heap	Insert	200000	3	0.005916
Heap	Insert	200000	4	0.005982
Heap	Insert	200000	5	0.005895
Heap	Insert	200000	Average	0.0059192
Heap	Insert	400000	1	0.011582
Heap	Insert	400000	2	0.011933
Heap	Insert	400000	3	0.01183
Heap	Insert	400000	4	0.011943
Heap	Insert	400000	5	0.011806
Heap	Insert	400000	Average	0.0118728

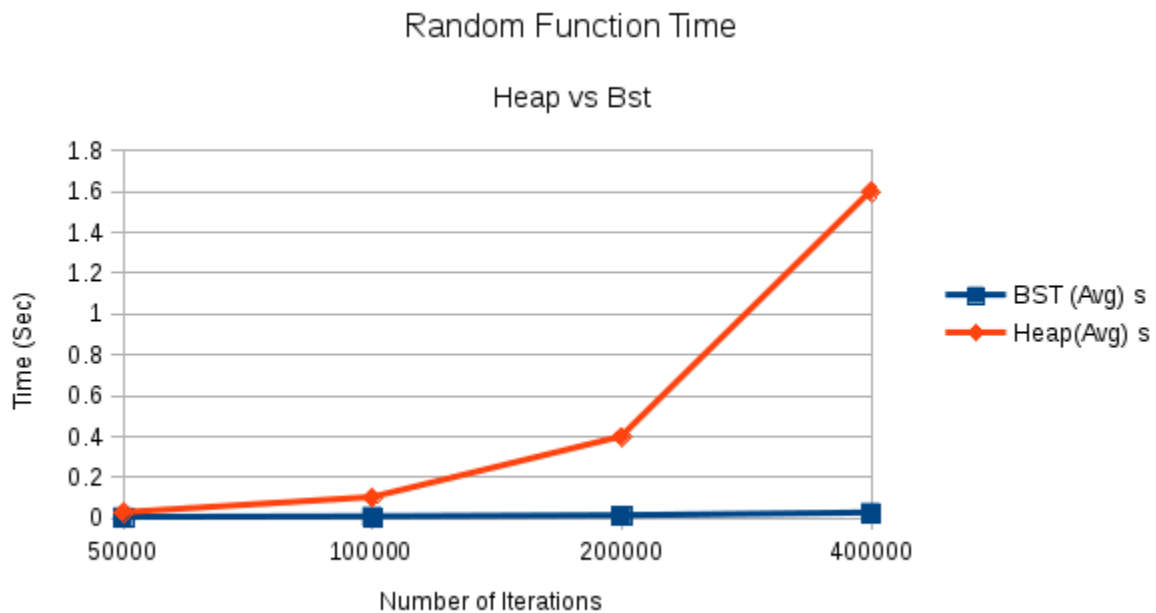
The table above compares the time it takes for a BST to insert versus the amount of time it takes a heap to insert. The heaps insert time is consistently across all trials about a thirtieth that of the BST. This is because when inserting in a BST the value traverses the entirety of one branch of the tree while the Heap simply has to insert at the back of an array and upheap- a much cheaper operation, as evidenced by the speed of heap insert. Remember that in this iteration we only performed  $n/10$  operations, which is the cause of the discrepancy between the above BST insert and the below BST Random. The corresponding chart is as follows:



The second part of the calculation was a bit tricky – I managed to fix the problems with my heap, but not the problems with my BST (removal specifically) so I compared the mixed functions of the heap with the insert function of the BST. I hope that I am not too terribly penalized for being unable to fix the BST because I already was in the last lab. Nonetheless, here is the cooresponding table:

Heap or BST	Operation	N	Trial	Time (sec)
BST	Random	50000	1	0.001975
BST	Random	50000	2	0.001989
BST	Random	50000	3	0.002047
BST	Random	50000	4	0.002002
BST	Random	50000	5	0.00197
BST	Random	50000	Average	0.0019966
BST	Random	100000	1	0.0046
BST	Random	100000	2	0.004593
BST	Random	100000	3	0.004624
BST	Random	100000	4	0.004538
BST	Random	100000	5	0.004652
BST	Random	200000	Average	0.0045014
BST	Random	200000	1	0.010328
BST	Random	200000	2	0.010503
BST	Random	200000	3	0.010349
BST	Random	200000	4	0.009807
BST	Random	200000	5	0.01019
BST	Random	200000	Average	0.0102354
BST	Random	400000	1	0.026059
BST	Random	400000	2	0.024316
BST	Random	400000	3	0.021882
BST	Random	400000	4	0.022382
BST	Random	400000	5	0.02353
BST	Random	400000	Average	0.0236338
Heap	Random	50000	1	0.025463
Heap	Random	50000	2	0.025125
Heap	Random	50000	3	0.023707
Heap	Random	50000	4	0.024441
Heap	Random	50000	5	0.024107
Heap	Random	50000	Average	0.0245686
Heap	Ranodm	100000	1	0.095253

Heap	Random	100000	2	0.0925
Heap	Random	100000	3	0.102521
Heap	Random	100000	4	0.10361
Heap	Random	100000	5	0.101872
Heap	Random	100000	Average	0.0991512
Heap	Random	200000	1	0.400816
Heap	Random	200000	2	0.406793
Heap	Random	200000	3	0.385864
Heap	Random	200000	4	0.39567
Heap	Random	200000	5	0.392811
Heap	Random	200000	Average	0.396391
Heap	Random	400000	1	1.62179
Heap	Random	400000	2	1.59489
Heap	Random	400000	3	1.60136
Heap	Random	400000	4	1.60079
Heap	Random	400000	5	1.57186
Heap	Random	400000	Average	1.59874



The above graph is of the averages of the times it takes to execute the functions. Obviously a direct comparison doesn't make sense as the BST only inserted while the heap performed four different functions randomly. The average time for the Heap is exponential because the larger the heap is, the longer the operations take, because the graph itself is larger.