

Assignment 5

By Shardul Negi

(001898181)

Problem Statement

You will perform more experiments to test that the depth/height of a Binary Search Tree after M (Hibbard) deletions and insertions will be proportional to the square root of N where N is the size of the tree when M is large. The consequence of this is that deletion, search and insertion will all end up being $O(N^{1/2})$ instead of $O(\lg N)$ which is what we would prefer.

Implementation Details

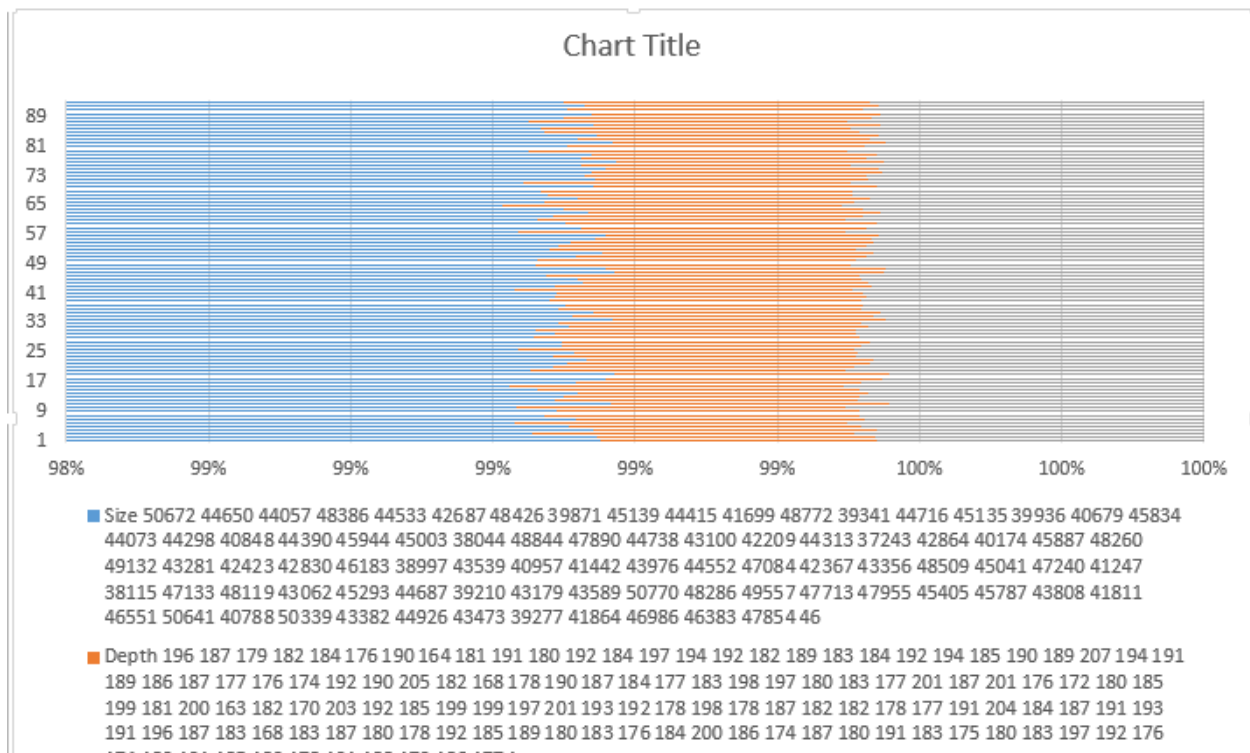
The main has been implemented for inserting and deleting X keys from the tree and checking the depth of the tree once created. Once done, I have used the following methods to calculate the result.

- 1: put(): will give the rand numbers to insert values to the tree
- 2: Delete() : will give rand numbers to delete values from the tree
- 3: size(): Will calculate the size of the tree. .

Experiments

On setting the value from 200(tree size) we insert and delete the key pair for the tree.

We will get the following relation in this.



Since Peach and Light blue is demonstrated by the Size and Root of N, we can see a clear relation between the two being equal. The console results are below and in the HTML file provided in the project.

Size of Tree	Depth	Square Root of N	Log of N to base 2
23289	163	152.60733927305068	14.507361072940473
21925	158	148.07092894960846	14.420289222219457
22853	171	151.17288737065187	14.480095945315428
21825	152	147.73286702694156	14.413694033404166
20792	150	144.19431334140748	14.343740918472282
25093	155	158.40770183296013	14.61409734238062
22287	159	149.28831166571615	14.443014811472094
24120	175	155.3061492665374	14.55794228678745
23953	150	154.76756766196203	14.547918737559977
21800	156	147.648230602334	14.412040514551652
20431	156	142.93704908105525	14.318472198389742
21987	158	148.28014027508877	14.42436314970957
20955	157	144.75041944425092	14.355006901017993
21959	155	148.18569431628683	14.422524735901288
22860	154	151.19523802024983	14.480537783101841
18691	138	136.71503209230505	14.190065137313467
23767	157	154.1654954910469	14.536672189507609
23352	154	152.81361195914454	14.51125849502268
23840	163	154.40207252495026	14.541096615349526
22759	157	150.86086304936745	14.474149548497307
22116	160	148.71449156917042	14.432082856351868
23152	168	152.15781281288187	14.498849206531725
23147	157	152.1413816159167	14.498537602599352
23734	158	154.05843047363555	14.534667644592403
25890	158	160.90369790654285	14.660107344787475
23480	162	153.23185047502363	14.519144787997416
24224	167	155.64061166674975	14.564149489985732
22888	159	151.28708519261954	14.482303782323182
21542	156	146.77193192160414	14.394864578184361
22853	164	151.17288737065187	14.480095945315428
22717	157	150.72159765607583	14.471484705109788
21971	159	148.22617852457776	14.423312914408891

As shown, the depth and the square root is coming to be almost equal in the 1000 tests that I ran for the tree.

Unit Test

Test Name	Duration (s)
testPut0	0.000
testPut1	0.000
testPut2	0.000
testPut3	0.000
testPutN	0.000
testDelete1	0.000
testDelete2	0.000
testDelete3	0.000
testDelete4	0.000
testPutAll	0.000
testTraverse	0.450
testSetRoot1	0.000
testSetRoot2	0.001

All the Junit test passed. One change has been made to the eQueue and Dequeue given by the professor.

Conclusion

The relation has been derived and mapped with respect to insert and deletion of the tree.