EECE 7205 HW5

Yaqiao Liu
NUID 001279512

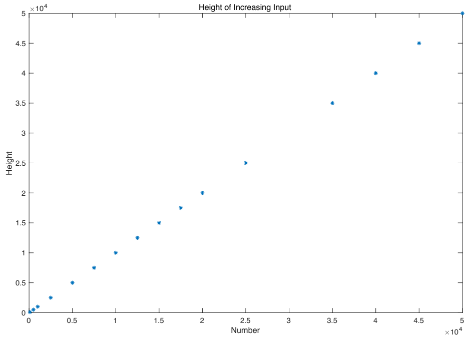
Question 2

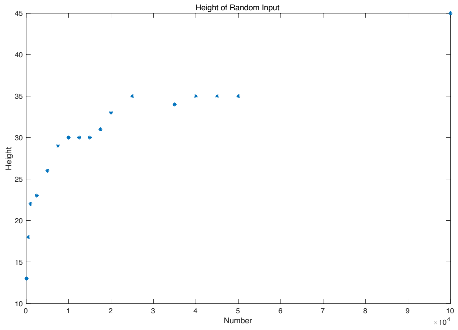
The raw data is attached in the end.

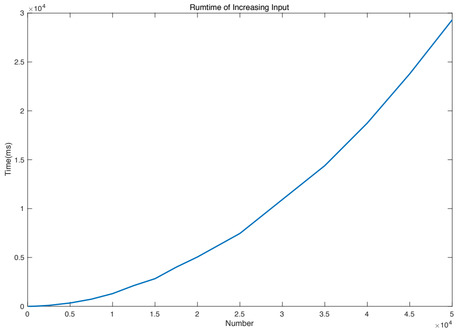
The images of height and runtime vary with numbers of node, and runtime varies with height are shown here.

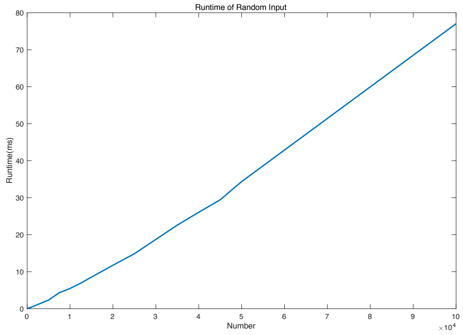
Increasing

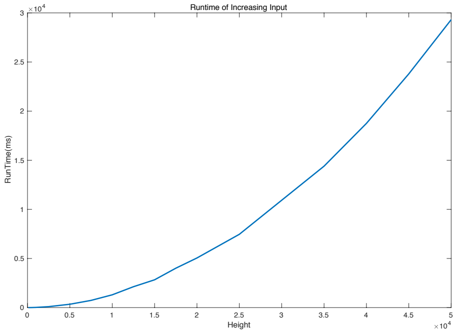
Random

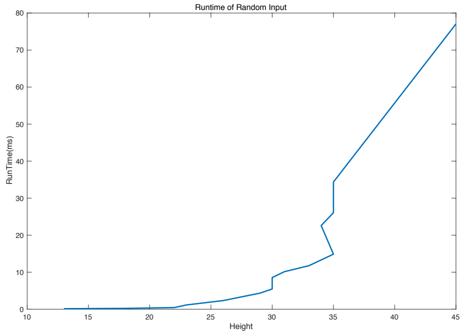












From the plot, we can see the obvious performance difference between these two programs.

As for height, the random main program has a lower height. The increasing program would create a one-side tree, which has a higher height.

As for runtime, the increasing program has a longer time. It is easy to understand. When the number of nodes increase, the height of the tree increases sharply. The time cost of each insert() function is O(h), when the h is large, the performance is bad. Furthermore, the height of tree is n-1, so the total time is the sum of a series of numbers which equal the number of current elements number – 1 in the tree. So the magnitude of total runtime can be expressed to O(n^2). The random program has a lower runtime, although it is not logn. It is because we cannot keep the whole tree as a completed binary tree. Because when the node number increases, the height of tree increases slowly, the total runtime is low.

Question 4

b) Here I can observe the height of the tree equals to the half of the input number. And the runtime decreases. The new implement can reduce the height of the tree and the runtime of the Insertion function.

The raw data:

increasing			random		balanced	
number	height	time(ms)	height	time(ms)	height	time(ms)
100	99	0.284	13	0.127	50	0.213
500	499	4.888	18	0.216	250	2.959
1000	999	17.872	22	0.435	500	10.65
2500	2499	95.925	23	1.144	1250	62.906
5000	4999	339.695	26	2.329	2500	209.303
7500	7499	731.469	29	4.333	3750	454.215
10000	9999	1299.12	30	5.456	5000	792.112
12500	12499	2123.32	30	6.915	6250	1236.81
15000	14999	2828.4	30	8.545	7500	1773.82
17500	17499	4006.78	31	10.136	8750	2378.02
20000	19999	5050.6	33	11.751	10000	3071.24
25000	24999	7459.65	35	14.864	12500	4699.14
35000	34999	14396.9	34	22.595	17500	8904.52
40000	39999	18750.7	35	26.063	20000	11563.7
45000	44999	23791.4	35	29.436	22500	14553
50000	49999	29308.1	35	34.358	25000	18783