Student Name:Guanhua Yang Tree Report

Student ID:301158515

n	Comparison of random insertion to search tree($C_{S.T}$)	$C_{S.T}/n$	$C_{S.T}/n\log_2 n$	$C_{S.T}/n^2$	Comparison of random insertion to avl tree($C_{A.T}$)	$C_{A.T}/n$	$C_{A.T}/n\log_2 n$	$C_{A.T}/n^2$	Comparison of random insertion to sorted search tree($C_{S.S.T}$)	$C_{S.S.T}/n$	$C_{S.S.T}/n\log_2 n$	$C_{S.S.T}/n^2$	Comparison of random insertion to sorted avl tree($C_{S.A.T}$)	$C_{S.A.T}/n$	$C_{S.A.T}/n\log_2 n$	$C_{S.A.T}/n^2$
5	6	1.2	0.516812	0.24	7	1.4	0.602947	0.28	10	2	0.861353	0.4	8	1.6	0.689082	0.32
10	22	2.2	0.662266	0.22	22	2.2	0.662266	0.22	45	4.5	1.354635	0.45	25	2.5	0.752575	0.25
20	72	3.6	0.832962	0.18	64	3.2	0.74041	0.16	190	9.5	2.198093	0.475	69	3.45	0.798255	0.1725
40	193	4.825	0.906626	0.120625	164	4.1	0.770397	0.1025	780	19.5	3.664086	0.4875	177	4.425	0.831466	0.110625
80	477	5.9625	0.943146	0.074531	400	5	0.790898	0.0625	3160	39.5	6.248094	0.49375	433	5.4125	0.856147	0.067656
160	1185	7.4063	1.011516	0.046289	942	5.8875	0.804091	0.036797	12720	79.5	10.8578	0.496875	1025	6.40625	0.87494	0.040039
320	2863	8.9469	1.075096	0.027959	2245	7.015625	0.843029	0.021924	51040	159.5	19.16623	0.498438	2369	7.403125	0.889593	0.023135
640	6459	10.09219	1.082629	0.015769	5167	8.073438	0.866069	0.012615	204480	319.5	34.27403	0.499219	5377	8.401563	0.901269	#DIV/0!
1280	14491	11.32109	1.0968	0.008845	11703	9.142969	0.885781	0.007143	818560	639.5	61.95548	0.499609	12033	9.400781	0.910758	0.007344
2560	32277	12.6082	1.113609	0.004925	25892	10.11406	0.893316	0.003951	3275520	1279.5	113.0108	0.499805	26625	10.40039	0.918606	0.004063
5120	71584	13.98125	1.134664	0.002731	56689	11.07207	0.898566	0.002163	13104640	2559.5	207.7191	0.499902	58369	11.4002	0.925196	0.002227
10240	157510	15.38184	1.154625	0.001502	124550	12.16309	0.913012	0.001188	52435680	5120.672	384.3792	0.500066	127977	12.49775	0.938134	0.00122
20480	343801	16.78716	1.17213	0.00082	268986	13.13408	0.917061	0.000641	209797960	10244.04	715.2697	0.500197	275433	13.44888	0.939041	0.000657

1.What is the observed complexity class of the `SearchTree` and `AVLTree` for the random data set? If a tree's performance doesn't clearly fall into one class, give a range.

For the random data set case of inserting random number into SearchTree, I observed that the complexity class is approximatly: O($n\log_2 n$) This is because, the ratio of comparison count of insert random number into search tree and nlog2n have the tendency to be about 1. As the size of the random sequence get bigger and bigger, the ratio appear to be constant roughly equal to 1. It is not exactly one is beacause there exist some variance or noise as our sequence is pick randomly. But for the other two case, n and n2, the ratio tends to grow faster for case n and slower for case n^2. Which means that these two case doesn't have the approximate true representation of the time complexity of the random insertion to search tree. It is as the size of the random sequence increase, the comparison of these case would not represent by the either case n or n2. Because the ratio is become either bigger or smaller very fast with repsect to its type.

For the random data set case of inserting random number into AVLTree, I observed that the complexity class is approximatly: O($n\log_2 n$) This is because, the ratio of comparison count of insert random number into AVL tree and nlog2n have the tendency to be about 1. As the size of the random sequence get bigger and bigger, the ratio appear to be constant roughly equal to 1. It is not exactly one is beacause there exist some variance or noise as our sequence is pick randomly. But for the other two case, n and n2, the ratio tends to grow faster for case n and slower for case n^2. Which means that these two case doesn't have the approximate true representation of the time complexity of the random insertion to AVL tree. It is as the size of the random sequence increase, the comparison of these case would not represent by the either case n or n2. Because the ratio is become either bigger or smaller very fast with repsect to its type.

2.What is the observed complexity class of the SearchTree and AVLTree for the sorted data set? Again, give a range if there is no clear single class.

For the sorted random data set case of the complexity of the SearchTree is approximatly O(n^2), This is because, the ratio of comparison count of insert sorted random number into search tree and n^2 have the tendency to close to 1 in compare with the two other case, n & $n\log_2 n$. For case n, the ratio grows extremly fast, it is about double for each of my defined n. And the same for the case $n\log_2 n$. It is not exactly one is beacause there exist some variance or noise as our sequence is pick randomly. But eventually the ratio would be equal to 1 for a very very large n. But for the two other cases, it doesn't have the approximate true representation of the time complexity of the sorted random insertion to search tree. It is as the size of the random sequence increase, the comparison of these casewould not represent by the either case n or $n\log_2 n$. Because the ratio is both become bigger.

For the sorted random data set case of the complexity of the AVLTree is approximatly O($n\log_2 n$), This is because, the ratio of comparison count of insert sorted random number into AVL tree and $n\log_2 n$ have the tendency to close to 1 in compare with the two case, n & n^2. For case n, the ratio grows fast, it become bigger and bigger for each of my defined n. And for the case n^2. It become small and smaller. Also for the case $n\log_2 n$, the ratio is not exactly one is beacause there exist some variance or noise as our sequence is pick randomly. But eventually the ratio would be equal to 1 for a very very large n. But for the two other cases, it doesn't have the approximate true representation of the time complexity of the sorted random insertion to AVL tree. It is as the size of the random sequence increase, the comparison of these case would not represent by either case n or n^2. Because the ratio is become either bigger or smaller very fast with repsect to its type.