The time for generating an AVL tree should be (n * log(n)), but my data shows that the complexity is actually much higher. Adding relatively few reports to a tree seems to greatly increase the time. I've gone over my algorithms, and it seems like it should be (n * log(n)), but the time for generating says otherwise. I tried generating trees from all the 2.5 million reports, but it took too long, well over 10 minutes and I was only a fraction of the way through, and it kept getting slower. Testing with a small sample showed my algorithm for finding accidents after a given date in a state appeared to work as intended.

As far as searching the AVL appears to be more efficient, as expected. It took only 33 microseconds to find 5260 reports after searching 25000 reports, but it to the treemap structure 310 microseconds to find 107 reports after searching 25000 reports. The more important number is the size of the tree searched, so 33/25000 is roughly 10 times faster than 310/25000.

Unless I made some serious mistake I could not recognize in my algorithm, the treemap data structure is much faster for generating trees. This makes sense, because the AVL structure requires constant manipulation when adding nodes, which gets more and more complex as the number increases.

Overall, the AVL is faster for searching than the treemap, but much, MUCH slower for generating the trees for the data. Assignment 7 data is on top, assignment 6 data is on the bottom.

reports placed	milliseconds to generate		reports found	nanoseconds to find
150000	149807		30047	110600
140000	77947		28099	112600
130000	42302		26036	99700
120000	25428		24114	107700
110000	16021		22146	94700
100000	9061		20258	81100
90000	5729		18181	88000
80000	3777		16222	74500
25000	542		5260	33900

milliseconds to generate vs. reports placed

nanoseconds to find vs. reports found
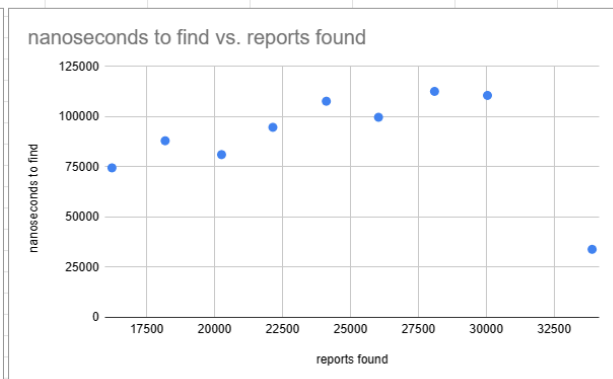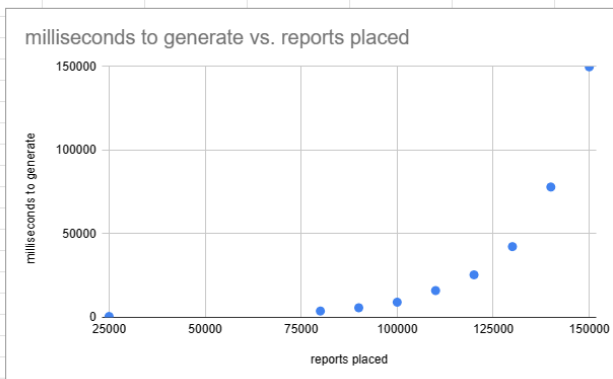
insert

find

microseconds	entries			microseconds	entries searched	entries found
8200660	2500000			563	2500000	105463
7675216	2250000			614	2250000	105463
7605555	2000000			647	2000000	105463
6273333	1750000			553	1750000	104385
5226805	1500000			731	1500000	82824
4365022	1250000			566	1250000	61534
3477747	1000000			543	1000000	40266
2515164	750000			582	750000	18891
1677752	500000			561	500000	107
801623	250000			459	250000	107
108673	25000			317	25000	107
30294	2500			310	2500	107
13506	250			309	250	74
8422	25			298	25	2
6776	1			344	1	0

entries vs. microseconds (insertion)

entries vs. microseconds (search)

1.27E-04*x + 377 R² = 0.645