Format:

Total rotation count

Average search depth

Total time

Ex:

```
10000
50
0.04s
```

	Sequential	Uniform	Skewed	Working Set
Bottom-Up	541425 5 0.040s	2542276 25 0.162s	2203264 22 0.095s	565610 5 0.058s
Top-Down	531961 5 0.072s	4309696 43 0.151s	3868680 38 0.134s	780544 7 0.076s
Semi-Splay	531872 5 0.182s	2933475 29 0.343s	2540346 25 0.249s	743681 7 0.190s
Weighted-Splay	431958 4 0.047s	3415782 34 0.133s	2894678 28 0.114s	799810 7 0.041s

1. They should not have an asymptotic bound difference in their complexity, since they both operate in the same time complexity. On the other hand, they do have one major difference, space. A bottom up approach requires parent pointers, n of them. On a large scale, this can cause problems with storage. A top down approach does take up a lot of memory while running because of recursion (if you have a really tall, tree, splaying a deep node can cause deep recursion), but it does require less space since it does not have parent pointers.

2. A splay tree is not thread safe due to the fact that basically any operation causes the tree to change. It needs to splay the node; therefore, they are not thread safe.

3. In any splay tree, the goal is not a balanced tree, but rather to bring the node to the top. A semi-splay tree brings the node halfway up the tree to remain somewhat balanced.

A weighted tree keeps frequently accessed nodes close to the top, so it is easier to access in the future.

4. Task 3 was all about the weighted splay tree. I think I liked making this implementation the most, just because it was one of the more intuitive ones. There was a little bit of trouble getting it working with search(), but I figured it out. The weights are useful in a situation like this (for splay trees) because they keep data that is more frequently accessed near the top. My implementation is basically a rewrite of the rotate functions. Each rotation checks if its parent has a greater value, if not, it moves up the tree. Each node starts with a 0 weight. That is it, all of the changes compared to a top down approach.