Alexander Sun – CS2223 – HW3

Collected Data

	HashTable (Avg over 100 iterations)	BST(Avg over 100 iterations)
Average Search Time Across TestSet(1/10 keys)	0.21 ms	0.55 ms

	HashTable (Avg over 100 iterations)	BST(Avg over 100 iterations)
Construction Time	212.24 ms	212.95 ms

Dealing with construction time first. For this sample size of around 1600 words, both took about the same amount of time to construct. This was a little surprising as a BST has to add a new Node, while the Hash table at its quickest just calculates a hash and sets a value. I originally thought the BST would be much slower but based on this size of 1600 words they are of similar speed.

The main differences come in the average search time. The Hash table operates at about double the speed of the BST, which isn't super surprising as the Hash Table has O(1) access time through a constant time hash and array lookup. The BST by comparison is a O(log(n)) at worst because it has to traverse and make comparison before it finds the value it's looking for (if present). This Hash Table is sized 4096 for a word count of 1600. Because of the sparse amount of words and low likelihood of collision (all terms are unique words), there is very little time spent linear probing so it is close to an O(1) and a very fast O(1) as hashing and looking up in an array are very quick.

This makes me conclude that the Hash Table is the best option for this task as its lookup time is empirically much shorter and about half of that of the BST. The construction time is also very equal based on my measurements so there's no real tradeoff there.

This being said, I think that if more collisions occurred and the Hash Table was more densely packed then the BST could win in terms of search time. However, I must wonder that construction of the BST in a much larger set will also take more time O(log n) insertion/addition vs. O(1) for the Hash Map. If the dataset was much much larger, I think the results would be different. The Hash Table also wastes a lot of memory, which is difficult to display or show.