Kern Espinoza
CS260
6 December, 2021

Assignment 5 Write-Up

3) It took **4453 ms** to build the index on my machine. Since insertion should generally be close to O(1), and since every single word must be considered even if it's already in the index, it should take about O(m) to build the entire index.

A) Self-Balancing BST insertion should be O(logn), so building the same index with that data structure should be O(m*logn). *m = 186,000 & n = 23,062*
   a) O(m) / O(m*logn) = 4453ms / t
   b) 186000 / 186000 * log2(23062) = 4453 / t
   c) t = 64538 ms = **64.53 seconds**
   d) The reason the Self-Balancing BST would take longer to build is because each insertion requires increasingly more traversal as the tree grows larger. A hash table involves traversal for probing when there are collisions, but the amount of probing stays the same as the backing array grows.

B) Array insertion should always be O(1), so it should take about O(m) to build, just like the hash map.
   a) O(m) / O(m) = 4453ms / t
   b) t = 4453 ms = **4.34 seconds**
   c) Although the O() is comparable, the array should be a bit faster to build, since it wouldn't do any hashing or probing for each new record. However, lookups will be around O(n/2) on the hypothetical array index, instead of around O(1) for the original hash map. So the array would take roughly about the same amount of time to build, but its lookups would be far slower.