Muangkong Yang
CSCD320 – HW4

1) Time Complexity of My Solver Tree

Without being able to pass in both dictionaries into the isWord method, I had to include building both dictionaries in my isWord method. *N,* being the number of digits in the sequence, building the dictionary with a TreeMap is O(logN) and with a Hashtable is O(N). My createTree method is called to build the solver tree. It has a time complexity of O(logN) since we are building the number of nodes based on N. Worst case scenario, we are adding 4 child members into each node. We have 4^N number of nodes with log(N) levels.

As of now my GRF = N(logN + logN) = N(2logN)

My method also calls findWord to search the Dictionary Trie for a word. This method has a time complexity of O(logN). I also call the containsKey method from the built-in Java. It has a time complexity of O(1).

So now my GRF = N(2logN + 1 + logN) = N(3logN + 1)

Dropping the constants and lower order terms, my time complexity for my method is

O(Nlog(N)).

2) Exhaustive Search

Also known as Brute-Force Search, checks for all possible solutions to satisfy a problem's statement. The part of the project that falls into this category of exhaustive search is when I check to see if each possible word is an actual word listed in the dictionary text document.

3) Branch and Bound Search

This search is a systematic method for solving optimization problems. Most of the candidate for solutions exist in the root of the problem. In the worst-case scenario, this may be slower than exhaustive search, but the average time is generally quite fast.

4) If we use a Prefix Tree to implement the dictionary, we can indeed use its property to accelerate the algorithm with Branch and Bound. To do that, we use the root node of the Prefix Tree to start. Then instead of going through all of the child members of the first node, we look at only the first letter of the sequence, traverse all the paths of that first letter in the dictionary, and print out all of the words when that word is a true word.

5) My Runs

```
------------------------TESTING isWord---------------------------
   Note: This program will only print out possible words
   from a sequence of numbers. If there is no actual words
   in the dictionary.txt file, then nothing will be printed.

Please enter your number sequence from any number from the range [2-9] (e.g. 2456789): 22
   Making Dictionary Trie...
   Making HashTable Dictionary ...
   Completed. Thanks for waiting!

   Now checking...
For your sequence of 22:
aa is a word in the Dictionary Trie!
aa is a word in the Hashtable Dictionary!
ab is a word in the Dictionary Trie!
ab is a word in the Hashtable Dictionary!
ac is a word in the Dictionary Trie!
ac is a word in the Hashtable Dictionary!
ba is a word in the Dictionary Trie!
ba is a word in the Hashtable Dictionary!
bb is a word in the Dictionary Trie!
bb is a word in the Hashtable Dictionary!
bc is a word in the Dictionary Trie!
bc is a word in the Hashtable Dictionary!
ca is a word in the Dictionary Trie!
ca is a word in the Hashtable Dictionary!
-----------------------------------------------------------------
```

```
------------------------TESTING isWord---------------------------
   Note: This program will only print out possible words
   from a sequence of numbers. If there is no actual words
   in the dictionary.txt file, then nothing will be printed.

Please enter your number sequence from any number from the range [2-9] (e.g. 2456789): 3
   Making Dictionary Trie...
   Making HashTable Dictionary ...
   Completed. Thanks for waiting!

   Now checking...
For your sequence of 3:
-----------------------------------------------------------------
```

```
------------------------TESTING isWord--------------------------
   Note: This program will only print out possible words
   from a sequence of numbers. If there is no actual words
   in the dictionary.txt file, then nothing will be printed.

Please enter your number sequence from any number from the range [2-9] (e.g. 2456789): 947266746
   Making Dictionary Trie...
   Making HashTable Dictionary ...
   Completed. Thanks for waiting!

   Now checking...
For your sequence of 947266746:
wisconsin is a word in the Dictionary Trie!
wisconsin is a word in the Hashtable Dictionary!
----------------------------------------------------------------
```

```
------------------------TESTING isWord--------------------------
   Note: This program will only print out possible words
   from a sequence of numbers. If there is no actual words
   in the dictionary.txt file, then nothing will be printed.

Please enter your number sequence from any number from the range [2-9] (e.g. 2456789): 7733428466
   Making Dictionary Trie...
   Making HashTable Dictionary ...
   Completed. Thanks for waiting!

   Now checking...
For your sequence of 7733428466:
prediction is a word in the Dictionary Trie!
prediction is a word in the Hashtable Dictionary!
----------------------------------------------------------------
```

```
------------------------TESTING isWord--------------------------
   Note: This program will only print out possible words
   from a sequence of numbers. If there is no actual words
   in the dictionary.txt file, then nothing will be printed.

Please enter your number sequence from any number from the range [2-9] (e.g. 2456789): 22263666368
   Making Dictionary Trie...
   Making HashTable Dictionary ...
   Completed. Thanks for waiting!

   Now checking...
For your sequence of 22263666368:
abandonment is a word in the Dictionary Trie!
abandonment is a word in the Hashtable Dictionary!
----------------------------------------------------------------
```

6) I worked with John Burtis on this homework and we did not attempt the extra credit portion.