ASSIGNMENT 1 SCRABBLE

TASK 1

This function finds the largest Anagram by searching through a sorted list and counting the adjacent elements until the next word adjacent to current word does not match then it will record the results
O(MN) Time complexity due to radix sort taking O(MN), O(N) coming from Counting sort.
O(MN) Space complexity used to store the dictionary.

TASK 2

This function is used to find anagram of a given query, done by sorting the characters in the query then the new sorted query is matched with an array of words (sorted by character) with the same length. When matches are found using a modified binary search that returns first occurrence in the array, the index is matched up with the original words and then appended to words list.
O(klogN + W) time complexity where W is the output size, log N is the time taken by binary search and K is the query length

TASK 3

This function is used to return all the words in the dictionary that can be made up with letters plus one wildcard tile. This was achieved by creating an array of the alphabet and adding each letter in the alphabet to the query word and sorting the characters in the query word and matching it up with other sorted words using getScrabbleWords()
O(klogN + W) time complexity where W is the output size, log N is the binary search and K is the query length