CPSC 319

Assignment#1

Alex Yeap

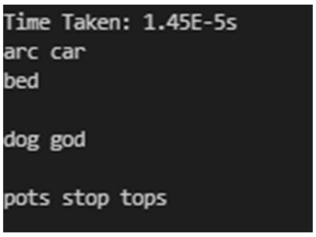
Alex.yeap@ucalgary.ca

Tutorial: T10

TA: Mobin Vahdati

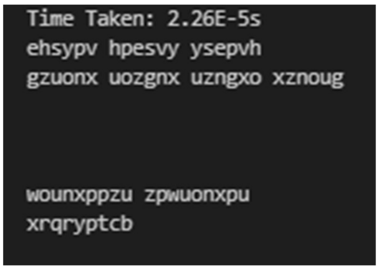
Time for each txt file:

Example_1.txt



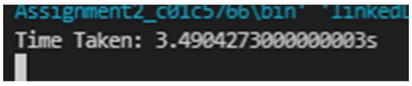
```
Time Taken: 1.45E-5s
arc car
bed

dog god

pots stop tops
```

Example_2.txt



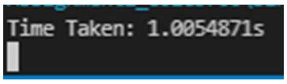
```
 Time Taken: 2.26E-5s
 ehsypv hpesvy ysepvh
 gzuonx uozgnx uzngxo xznoug



 wounxppzu zpwuonxpu
 xrqryptcb
```

Example_3.txt

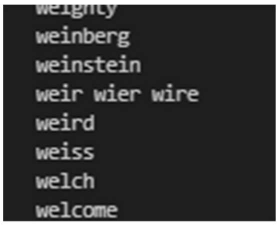


```
 Assignment2_c01c5766\bin' 'linkedL
 Time Taken: 3.4904273000000003s
```

Small.txt

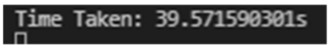


```
Time Taken: 1.0054871s
```



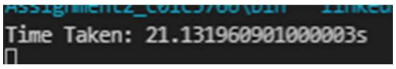
```
   weighty
   weinberg
   weinstein
   weir wier wire
   weird
   weiss
   welch
   welcome
```

Medium.txt



```
 Time Taken: 39.571590301s
```

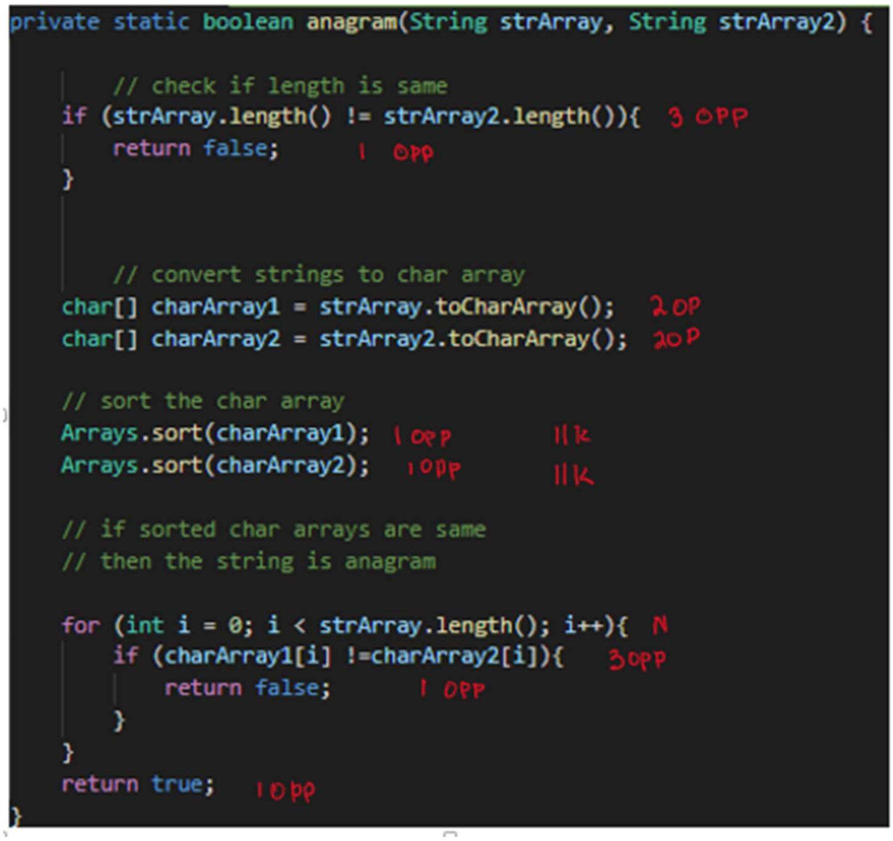
Large.txt



```
Assignment2_c01c5766\bin' 'linked
Time Taken: 21.131960901000003s
```

Complexity Analysis

1. **Big-Oh Complexity: (Worst Case)**



```
private static boolean anagram(String strArray, String strArray2) {

        // check if length is same
    if (strArray.length() != strArray2.length()){
        return false;
    }


        // convert strings to char array
    char[] charArray1 = strArray.toCharArray();
    char[] charArray2 = strArray2.toCharArray();

    // sort the char array
    Arrays.sort(charArray1);
    Arrays.sort(charArray2);

    // if sorted char arrays are same
    // then the string is anagram

    for (int i = 0; i < strArray.length(); i++){
        if (charArray1[i] !=charArray2[i]){
            return false;
        }
    }
    return true;
}
```

3+1+2+2+1+1+N(3+1)+1 = 4N + 11 = 0(N)

22k represent comparing to words with the length of 11 letters.

The worst-case scenario would be comparing 2 words with the largest length in the array. It would take more time to compare each letter which will result in a longer time

2. **The worst-case complexity when checking if 2 words are anagrams would take place by using medium.txt file. N is equal to 105989 for the input list or words and L is equal to 9 as it increases to the maximum of 9 letters. When we compare to the large.txt file, the large.txt file**

varies in length of each word when the medium.txt file gradually increases in length. This results in worst case complexity as we have a higher time (39s) compared to the time of large.txt (21s). running the program will have a total longer time due to the methods used in main and all the sorting algorithms involved. The larger the txt file, the longer it will take to sort.