Assignment 7

Name : Hrishikesh Rajan
Email : hrishikeshrajan3@gmail.com

Q.3) An e-commerce site tracks the purchases made each day. The product that is purchased
the most one day is the featured product for the following day. If there is a tie for the product
purchased most frequently, those product names are ordered alphabetically ascending and
the last name in the list is chosen.[Amazon]
['yellowShirt', 'redHat', 'blackShirt', 'bluePants', 'redHat','pinkHat', 'blackShirt', 'yellowShirt',
'greenPants', 'greenPants', 'greenPants']
'yellowShirt' - 2
'redHat' - 2
'blackShirt' - 2
'bluePants' - 1
'greenPants' - 3
'pinkHat' - 1
Output - greenPants

CODE

```
function shop(arr){


   arr.sort()
   console.log(arr)
   const map = {}

   let max = 0;
   let products = []

   for (const item of arr) {
       map[item] = (map[item] || 0)+1
```

```
    }

    for (const key in map) {
            if(max < map[key]){
                max = map[key]
            }

    }

    for ([key,values] of Object.entries(map)){

    if(max == values ){
        products.push(key)
    }
    }

    return products[products.length-1]

}
```

INPUT

```
const arr = ['yellowShirt', 'redHat', 'blackShirt', 'bluePants',
'redHat','pinkHat', 'blackShirt', 'yellowShirt',
'greenPants', 'greenPants','greenPants']
const result = shop(arr)
console.log(result)
```

OUTPUT

```
greenPants
```

ANALYSIS

Here we do four steps to reach the final output
First we sort the input array which rearranges the words in ascending alphabetical order.Then we perform a mapping which counts the frequency of each word in the array.That will help to also remove the duplicate elements as well.Then we search for the max value of values in map and store to a variable.On the next step we again loop through that and compare the values with max and if it matches we push that to new array and since the array is sorted we lony get correct output at the end of the array (specified in the question)

EXPLANATION:

Time complexity of the sorting takes 0 (n ^2), since that heavily depend on the language what use sorting in behind

Time complexity for mapping the key-value pairs = O (n), Single loop iterating upto n times

Time complexity for updating the max value = O (n), Single loop iterating upto n times

Time complexity for converting the object into array = O (n), Single loop iterating upto n times

Total Time Complexity = O (n ^ 2)+ O (n) + O (n) + O (n)
= O (n^2)
Total Space Complexity = O (n), here we use an extra array to store the result;