GTU Department of Computer Engineering

CSE 222/505 - Spring 2023

Homework 6 Report

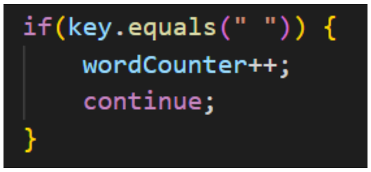
BERRU LAFCI

1901042681

myMap

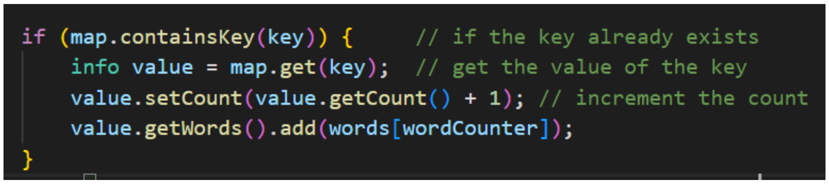
While building the map:

- After process the string, it initializes a variable wordCounter to keep track of the current word index and splits the preprocessedStr into an array of words using the space (" ") as the delimiter.



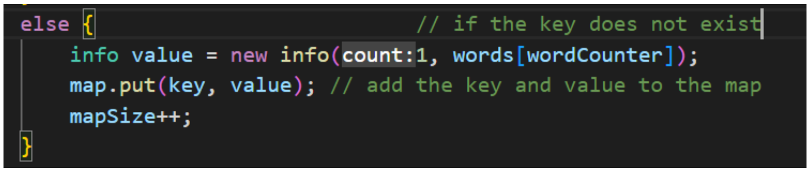
```
if(key.equals(" ")) {
    wordCounter++;
    continue;
}
```

It traverse through the string. If the map already contains the current key, it adds the current word to the list of words and increase the count in the **info** object.



```
if (map.containsKey(key)) {     // if the key already exists
    info value = map.get(key);  // get the value of the key
    value.setCount(value.getCount() + 1); // increment the count
    value.getWords().add(words[wordCounter]);
}
```

If the map does not contain the current key, it creates a new **info** object with a count of 1 and the current word.

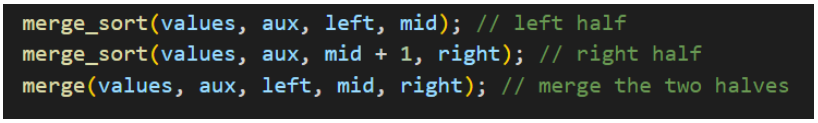


```
else {                          // if the key does not exist
    info value = new info(count:1, words[wordCounter]);
    map.put(key, value); // add the key and value to the map
    mapSize++;
}
```

mergeSort

The merge_sort() method is called with the values array, aux array, and the indices representing the leftmost and rightmost elements of the array. It recursively divides the array into smaller halves until the left index is less than the right index.

The merge_sort() method then calls itself twice, once for the left half of the array (from left to mid) and once for the right half of the array (from mid + 1 to right).



```
merge_sort(values, aux, left, mid); // left half
merge_sort(values, aux, mid + 1, right); // right half
merge(values, aux, left, mid, right); // merge the two halves
```

The actual mergeSort algorithm of the two sorted subarrays is performed by the merge() method:

The merge() method creates a temporary array (temp) to store the merged values and a temporary ArrayList to store the corresponding keys. It compares the values from the left and right subarrays and copies them to the temp array and tempKeys ArrayList in **sorted order.** If there are any remaining elements in either the left or right subarray, they are copied to the temp array and tempKeys ArrayList.

Once the merging process is completed, the sortMap() method creates a new sortedMap and copies the sorted values and corresponding keys from the original map to the sortedMap.

In summary, the merge sort algorithm is implemented by recursively dividing the array into smaller halves, sorting them separately, and then merging the sorted halves back together. This process continues until the entire array is sorted.

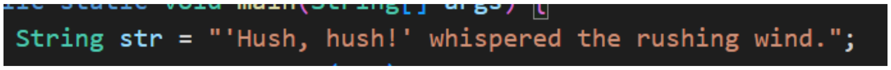
The time complexity of the merge_sort method:

The function divides the array in half each time it is called, resulting in a **binary tree of recursive calls.** Therefore, the depth of the recursion tree is logarithmic to the size of the input array, given by log(n), where n is the number of elements in the array.

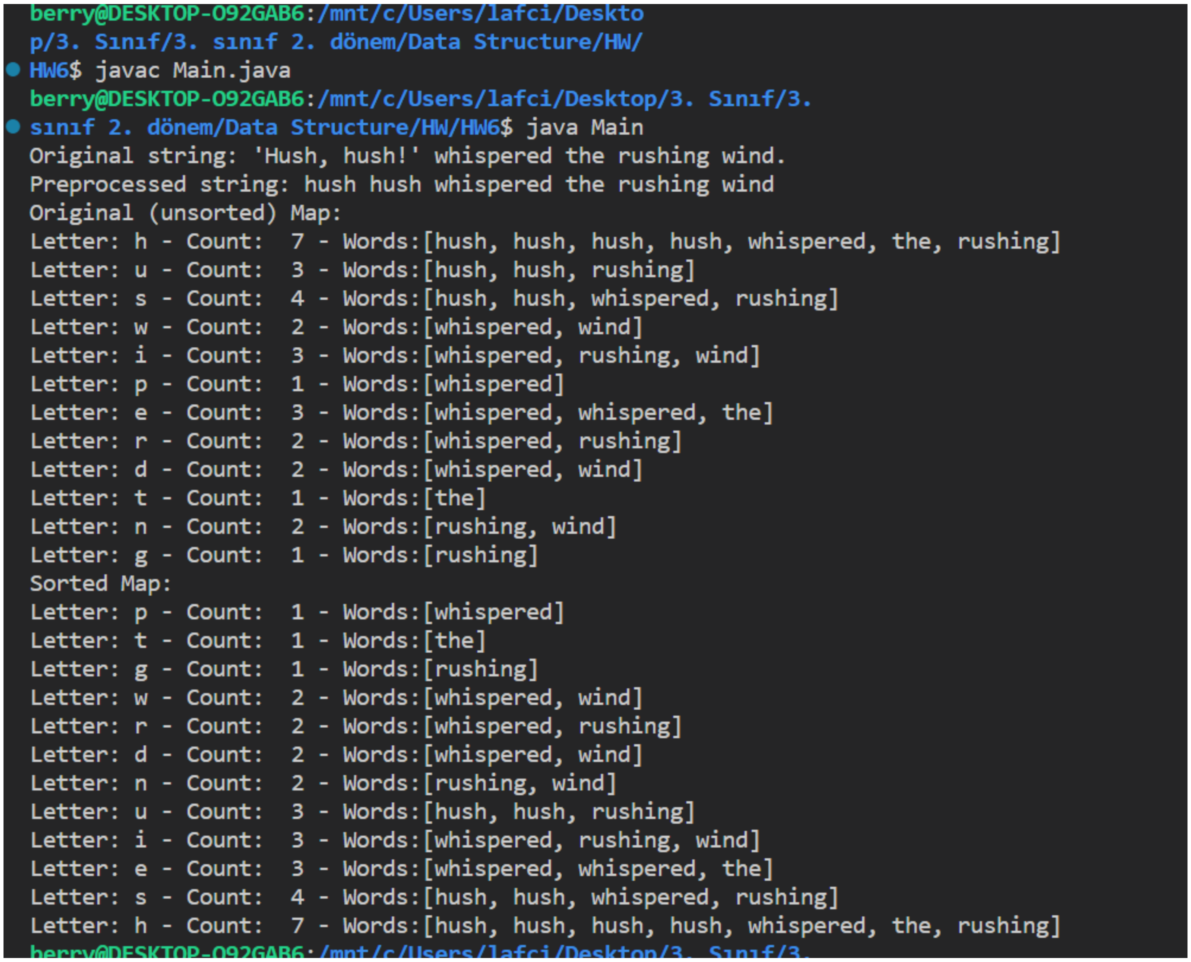
At each level of the recursion, the function performs a merge operation, which takes linear time proportional to the size of the merged subarray. Since the size of the merged subarray doubles at each level, the total time complexity of merging all subarrays is also linear to the size of the input array.

Therefore, the overall time complexity of the merge_sort function is **O(n log(n)),** where n is the number of elements in the input array.

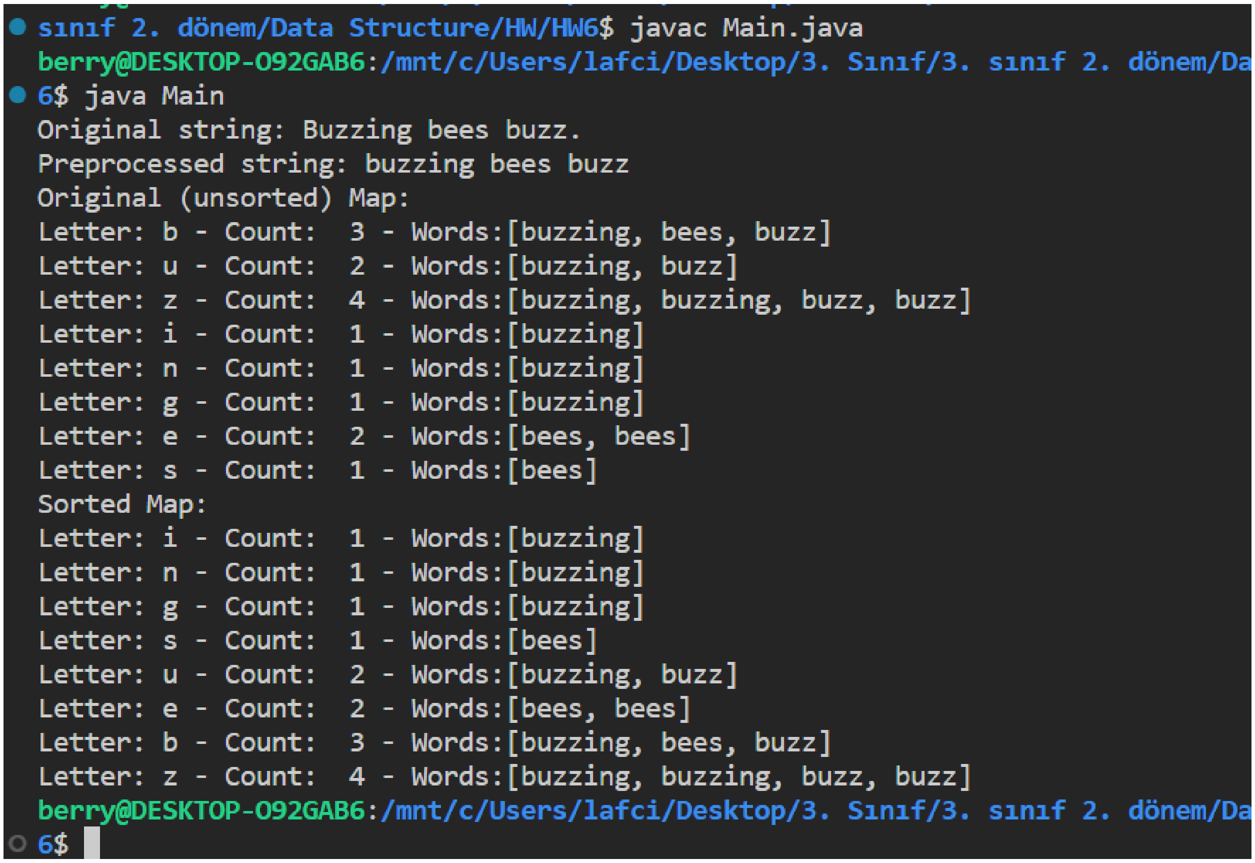
Test Cases:



```java
String str = "'Hush, hush!' whispered the rushing wind.";
```

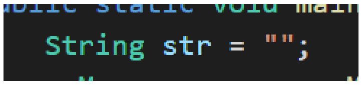


```
berry@DESKTOP-O92GAB6:/mnt/c/Users/lafci/Deskto
p/3. Sınıf/3. sınıf 2. dönem/Data Structure/HW/
HW6$ javac Main.java
berry@DESKTOP-O92GAB6:/mnt/c/Users/lafci/Desktop/3. Sınıf/3.
sınıf 2. dönem/Data Structure/HW/HW6$ java Main
Original string: 'Hush, hush!' whispered the rushing wind.
Preprocessed string: hush hush whispered the rushing wind
Original (unsorted) Map:
Letter: h - Count:  7 - Words:[hush, hush, hush, hush, whispered, the, rushing]
Letter: u - Count:  3 - Words:[hush, hush, rushing]
Letter: s - Count:  4 - Words:[hush, hush, whispered, rushing]
Letter: w - Count:  2 - Words:[whispered, wind]
Letter: i - Count:  3 - Words:[whispered, rushing, wind]
Letter: p - Count:  1 - Words:[whispered]
Letter: e - Count:  3 - Words:[whispered, whispered, the]
Letter: r - Count:  2 - Words:[whispered, rushing]
Letter: d - Count:  2 - Words:[whispered, wind]
Letter: t - Count:  1 - Words:[the]
Letter: n - Count:  2 - Words:[rushing, wind]
Letter: g - Count:  1 - Words:[rushing]
Sorted Map:
Letter: p - Count:  1 - Words:[whispered]
Letter: t - Count:  1 - Words:[the]
Letter: g - Count:  1 - Words:[rushing]
Letter: w - Count:  2 - Words:[whispered, wind]
Letter: r - Count:  2 - Words:[whispered, rushing]
Letter: d - Count:  2 - Words:[whispered, wind]
Letter: n - Count:  2 - Words:[rushing, wind]
Letter: u - Count:  3 - Words:[hush, hush, rushing]
Letter: i - Count:  3 - Words:[whispered, rushing, wind]
Letter: e - Count:  3 - Words:[whispered, whispered, the]
Letter: s - Count:  4 - Words:[hush, hush, whispered, rushing]
Letter: h - Count:  7 - Words:[hush, hush, hush, hush, whispered, the, rushing]
berry@DESKTOP-O92GAB6:/mnt/c/Users/lafci/Desktop/3. Sınıf/3.
```

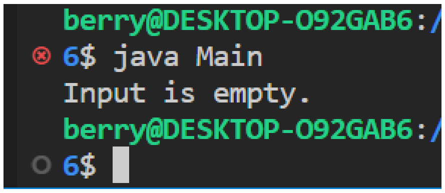


```
● sınıf 2. dönem/Data Structure/HW/HW6$ javac Main.java
  berry@DESKTOP-O92GAB6:/mnt/c/Users/lafci/Desktop/3. Sınıf/3. sınıf 2. dönem/Da
● 6$ java Main
  Original string: Buzzing bees buzz.
  Preprocessed string: buzzing bees buzz
  Original (unsorted) Map:
  Letter: b - Count:  3 - Words:[buzzing, bees, buzz]
  Letter: u - Count:  2 - Words:[buzzing, buzz]
  Letter: z - Count:  4 - Words:[buzzing, buzzing, buzz, buzz]
  Letter: i - Count:  1 - Words:[buzzing]
  Letter: n - Count:  1 - Words:[buzzing]
  Letter: g - Count:  1 - Words:[buzzing]
  Letter: e - Count:  2 - Words:[bees, bees]
  Letter: s - Count:  1 - Words:[bees]
  Sorted Map:
  Letter: i - Count:  1 - Words:[buzzing]
  Letter: n - Count:  1 - Words:[buzzing]
  Letter: g - Count:  1 - Words:[buzzing]
  Letter: s - Count:  1 - Words:[bees]
  Letter: u - Count:  2 - Words:[buzzing, buzz]
  Letter: e - Count:  2 - Words:[bees, bees]
  Letter: b - Count:  3 - Words:[buzzing, bees, buzz]
  Letter: z - Count:  4 - Words:[buzzing, buzzing, buzz, buzz]
  berry@DESKTOP-O92GAB6:/mnt/c/Users/lafci/Desktop/3. Sınıf/3. sınıf 2. dönem/Da
○ 6$
```

Empty input (with exception in myMap constructor):



```
ublic static void main
  String str = "";
```



```
  berry@DESKTOP-O92GAB6:/
⊗ 6$ java Main
  Input is empty.
  berry@DESKTOP-O92GAB6:/
○ 6$
```