GIT Department of Computer Engineering
CSE 222/505 - Spring 2023
Homework # Report

Yunus KARA

-

1. SYSTEM REQUIREMENTS

In this project, We are wanted to implement search algorithms which are Selection sort, Insertion Sort, Bubble Sort and Quick sort. My design is ascending order. We used our code which was coded in hw6. We added merge sort in that code.

2. PROBLEM SOLUTION APPROACH

In quick sort algorithm last element was selected as a pivot. Then it starts comparing and changing. There 3 different cases. Best cases Average cases and Worst Cases. I gave 3 different input for each cases. And output is going to be below. And due to the fact that we used our hw6 code, input validation. It just copied then pasted.

3. Test CASES

- **Bubble sort:**

  Compare to the other algorithms. When it has small size then it takes less than even if it has O(n^2) complexity.

  Best Case: O(n) which is already sorted. There is a Boolean variable that's controlling.

  Worst Case: O(n^2) which is sorted by oppositely. If you want to sorted it in ascending order, and your input is decreasing order. It will take time.

  Average: O(n^2) will take time. It is because nested loops.

- **Selection Sort:**
  - **Best Case:** If it is already sorted. It will take less time due to swapping operations. O(n^2).
  - **Worst Case:** It will take more time due to moving cursor every time to end. When input is in descending order and you want to sorted it ascending order. O(n^2).
  - **Average:** It will take O(n^2) when input randomly in map structure.
- **Insertion Sort:**
  - **Best Case:** It will take O(n). Because when it was sorted due to inner loop is not going to work.
  - **Worst Case:** is in ascending order, and you want to sort it in descending order. It will take O(n^2)
  - **Average Case:** It will take O(n^2) when input randomly in the list.
- **Quick Sort:**

- **Best Case:** When pivot is the middle element which will be middle element in sorted array. Then it will take less time O(nlogn).
- **Worst Case:** O(n^2) when your pivot is the smallest element while ascending ordering. Or, your pivot is the greatest element while descending order.
- **Average:** O(nlogn) When pivot is not smallest or greatest element and not middle element

- **Merge Sort:**

  It is always O(nlogn). It splits into subarrays which causes logn then it merge them O(n). Even if it is sorted it will nlogn time but it will may do less operation compare to unsorted array.

To conclusion, if you have a large data and you don't care space then you should use merge sort algorithm. If you have a large data and you are trouble with space then you should use quicksort algorithm. In my option with small data set for sorting bubble sort or insertion sort are the best option for coding and for speed and memory.

In HW6, it was highlighted that the ordering of the letters with same count value should be the same as their addition order to myMap object. However, for these 4 sorting algorithms, the case might not be the same. For example quick sort doesnt keep the order of the same count value. The others are the same for order of the same count value.

Measure times are in the Result section.

4. RUNNING AND RESULTS

Quick <Insertion <=bubble <selection < merge sort this home
Takes more time!

BUBLE SORT:

```
Original string is a bb ccc dddd eeeee ffffff gggggg hhhhhhh iiiiiiii xxxxxxxxx
Preprocessed string is a bb ccc dddd eeeee ffffff gggggg hhhhhhh iiiiiiii xxxxxxxxx
{}
--------
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiii
iiiii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxx
xx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiii
iiiii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxx
xx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx]
}
3 millisecond time required
 BUBLE SORT!!!!!!!!!!!!!!!!!! best case is done!!!!
Preprocessed string is xxxxxxxxx iiiiiiiii hhhhhhh ggggggg ffffff eeeee dddd ccc bb a
{}
--------
{x=Counted =10  [xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxx
x, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, b=Counted =2  [bb, bb]
, a=Counted =1  [a]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
```

```
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
7 millisecond time required
 BUBLE SORT!!!! Worst Case is done!!!!
Preprocessed string is eeeee ffffff a hhhhhhhh dddd ccc iiiiiiiii bb xxxxxxxxx
{}
--------
{e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, a=Counted =1  [a]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, b=Counted =2  [bb, bb]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
[a, b, c, d, e, f, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
4 millisecond time required
BUBLE SORT------------------Avarage case is don------------
```

Insertion Sort:

```
Original string is a bb ccc dddd eeeee ffffff ggggggg hhhhhhhh iiiiiiiii xxxxxxxxx
Preprocessed string is a bb ccc dddd eeeee ffffff ggggggg hhhhhhhh iiiiiiiii xxxxxxxxx
{}
--------
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii,
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx,
 xxxxxxxxx, xxxxxxxxx, xxxxxxxxx]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii,
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx,
 xxxxxxxxx, xxxxxxxxx, xxxxxxxxx]
}
3 millisecond time required
 INSERTION SORT--!!!!!!!!!!!!!!!!!! best case is done!!!!
Preprocessed string is xxxxxxxxx iiiiiiiii hhhhhhhh ggggggg fffffff eeeee dddd ccc bb a
{}
--------
{x=Counted =10  [xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx,
xxxxxxxxx, xxxxxxxxx, xxxxxxxxx]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii,
ii, iiiiiiiii]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, b=Counted =2  [bb, bb]
, a=Counted =1  [a]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
```

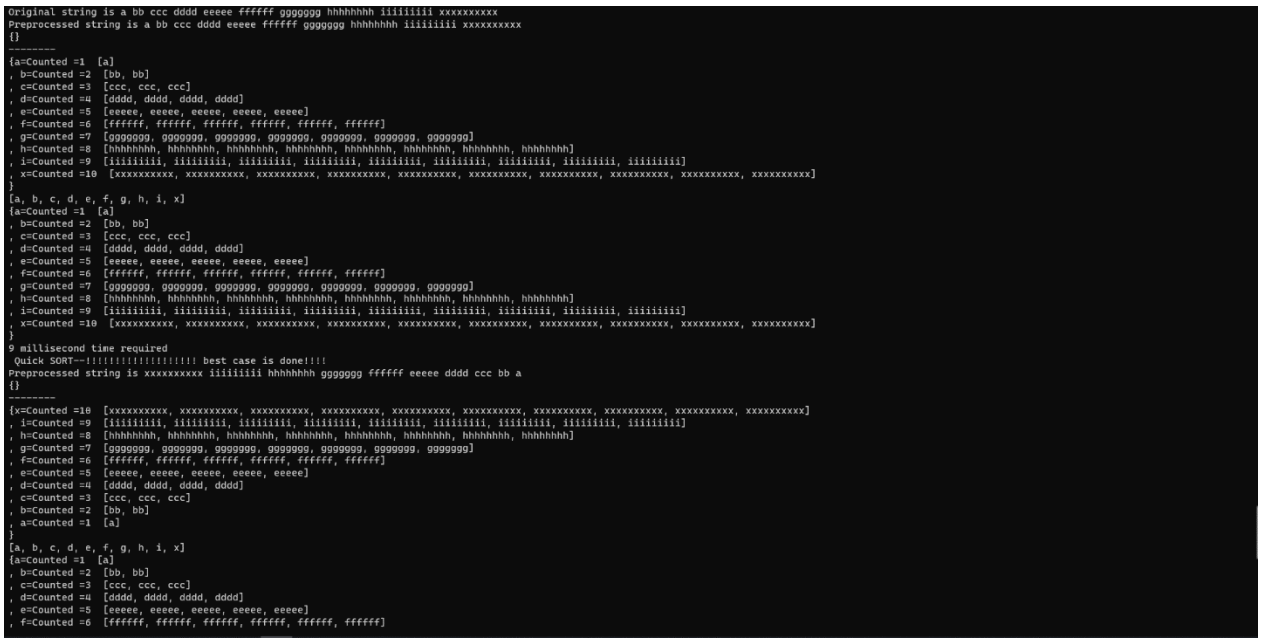
```
xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, b=Counted =2  [bb, bb]
, a=Counted =1  [a]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
6 millisecond time required
 INSERTION SORT--!!!! Worst Case is done!!!!
Preprocessed string is eeeee ffffff a hhhhhhhh dddd ccc iiiiiiiii bb xxxxxxxxxx
{}
--------
{e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, a=Counted =1  [a]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, b=Counted =2  [bb, bb]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
[a, b, c, d, e, f, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
5 millisecond time required
INSERTION SORT----------------------Avarage case is don-------------
```

Selection SORT:

```
Original string is a bb ccc dddd eeeee ffffff ggggggg hhhhhhhh iiiiiiiii xxxxxxxxx
Preprocessed string is a bb ccc dddd eeeee ffffff ggggggg hhhhhhhh iiiiiiiii xxxxxxxxx
{}
--------
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx,
 xxxxxxxxx, xxxxxxxxx, xxxxxxxxx]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx,
 xxxxxxxxx, xxxxxxxxx, xxxxxxxxx]
}
3 millisecond time required
 SELECTION SORT--!!!!!!!!!!!!!!!!!! best case is done!!!!
Preprocessed string is xxxxxxxxx iiiiiiii hhhhhhhh ggggggg ffffff eeeee dddd ccc bb a
{}
--------
{x=Counted =10  [xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx, xxxxxxxxx,
xxxxxxxxx, xxxxxxxxx, xxxxxxxxx]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiii
ii, iiiiiiiii]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, b=Counted =2  [bb, bb]
, a=Counted =1  [a]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
```

```
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiii
ii, iiiiiiiii]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, b=Counted =2  [bb, bb]
, a=Counted =1  [a]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
6 millisecond time required
 SELECTION SORT--!!!! Worst Case is done!!!!
Preprocessed string is eeeee ffffff a hhhhhhhh dddd ccc iiiiiiiii bb xxxxxxxxx
{}
--------
{e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, a=Counted =1  [a]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiii
ii, iiiiiiiii]
, b=Counted =2  [bb, bb]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
[a, b, c, d, e, f, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
4 millisecond time required
SELECTION SORT----------Avarage case is don----------
```

Quick SORT




```
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, b=Counted =2  [bb, bb]
, a=Counted =1  [a]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
8 millisecond time required
 Quick SORT--!!!! Worst Case is done!!!!
Preprocessed string is eeeee ffffff a hhhhhhhh dddd ccc iiiiiiiii bb xxxxxxxxx
{}
--------
{e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, a=Counted =1  [a]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii]
, b=Counted =2  [bb, bb]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
[a, b, c, d, e, f, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
2 millisecond time required
Quick SORT----------------------Avarage case is don-------------
```

MERGE SORT:

```
Original string is a bb ccc dddd eeeee ffffff gggggg hhhhhhh iiiiiiiii xxxxxxxxx
Preprocessed string is a bb ccc dddd eeeee ffffff gggggg hhhhhhh iiiiiiiii xxxxxxxxx
{}
--------
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
5 millisecond time required
 MERGE SORT--!!!!!!!!!!!!!!!!!! best case is done!!!!
Preprocessed string is xxxxxxxxx iiiiiiiii hhhhhhh ggggggg ffffff eeeee dddd ccc bb a
{}
--------
{x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, b=Counted =2  [bb, bb]
, a=Counted =1  [a]
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
```

```
}
[a, b, c, d, e, f, g, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, g=Counted =7  [ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg, ggggggg]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
8 millisecond time required
 MERGE SORT--!!!! Worst Case is done!!!!
Preprocessed string is eeeee ffffff a hhhhhhhh dddd ccc iiiiiiiii bb xxxxxxxxx
{}
--------
{e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, a=Counted =1  [a]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, c=Counted =3  [ccc, ccc, ccc]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, b=Counted =2  [bb, bb]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
[a, b, c, d, e, f, h, i, x]
{a=Counted =1  [a]
, b=Counted =2  [bb, bb]
, c=Counted =3  [ccc, ccc, ccc]
, d=Counted =4  [dddd, dddd, dddd, dddd]
, e=Counted =5  [eeeee, eeeee, eeeee, eeeee, eeeee]
, f=Counted =6  [ffffff, ffffff, ffffff, ffffff, ffffff, ffffff]
, h=Counted =8  [hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh, hhhhhhhh]
, i=Counted =9  [iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiiiii, iiiiiii
ii, iiiiiiiii]
, x=Counted =10  [xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx,
 xxxxxxxxxx, xxxxxxxxxx, xxxxxxxxxx]
}
7 millisecond time required
MERGE SORT--------------------Avarage case is don-------------
PS C:\Users\Yunus\Desktop\data\hw7>
```