

#1,2,3

'''

Solved in class

'''

#4

def SortString(s):

""" s is assumed to be a string of lowercase letters,

rerurns sorted s"""

#string to list

L=list(s)

#print(l) # checkpoint

Creating dictionaries letters-numbers

lToN={}

nToI={}

letters="abcdefghijklmnopqrstuvwxyz"

numbers="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26"

Numbers=numbers.split(",")

for i in range (26):

lToN[letters[i]]=int(Numbers[i])

nToI[Numbers[i]]=letters[i]

#letters to numbers

for i in range (len(L)):

temp=L[i]

L[i]=lToN[temp]

#print(L) #checkpoint

Sorted_L=merge_sort(L)

numbers to letters

for i in range (len(L)):

temp=Sorted_L[i]

Sorted_L[i]=nToI[str(temp)]

```
return (''.join((Sorted_L)))
```

```
def merge(left, right):  
    result = []  
    i,j = 0,0  
    while i < len(left) and j < len(right):  
        if left[i] < right[j]:  
            result.append(left[i])  
            i += 1  
        else:  
            result.append(right[j])  
            j += 1  
    while (i < len(left)):  
        result.append(left[i])  
        i += 1  
    while (j < len(right)):  
        result.append(right[j])  
        j += 1  
    #print('merge: ' + str(left) + '&' + str(right) + ' to ' +str(result))  
    return result
```

```
def merge_sort(L):  
    # print('merge sort: ' + str(L))  
    if len(L) < 2:  
        return L[:]  
    else:  
        middle = len(L)//2  
        left = merge_sort(L[:middle])  
        right = merge_sort(L[middle:])
```

```
return merge(left, right)
```

```
print(\
```

```
SortString("ffoasjgzdhthffxyjsyznynjzrthhtfzfgnnfxyjxfhzzrhnrtaajryktulipuiproihdzjxhkcgigjrr  
ihh"))
```

```
'''
```

Q:

What is complexity of the code above?

A:

We assume that accessing a dictionary is $O(n)$, so the loop it is nested in is $O(n^2)$.

Merge_sort

is $O(n \log(n))$, so the total complexity on $O(n^2)$.

```
'''
```

```
'''
```

#5 from <https://www.geeksforgeeks.org/python-program-for-insertion-sort/>

Python program for implementation of Insertion Sort

This code is contributed by Mohit Kumra

Function to do insertion sort

```
def insertionSort(arr):
```

```
    # Traverse through 1 to len(arr)
```

```
    for i in range(1, len(arr)):
```

```
        key = arr[i]
```

```
        # Move elements of arr[0..i-1], that are
```

```
        # greater than key, to one position ahead
```

```
        # of their current position
```

```
        j = i-1
```

```
        while j >= 0 and key < arr[j] :
```

```
        arr[j+1] = arr[j]
        j -= 1
    arr[j+1] = key
```

```
# Driver code to test above
```

```
arr = [12, 11, 13, 5, 6]
```

```
insertionSort(arr)
```

```
print ("Sorted array is:")
```

```
for i in range(len(arr)):
```

```
    print ("%d" %arr[i])
```

```
'''
```