Ex no: 2
12-03-2023

Name: M.Rohith
3122 21 5001 085

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam - 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)

UCS2403: DESIGN & ANALYSIS OF ALGORITHMS

Assignment 2

1. (a) Write a program to find unique (non-repeating) elements in a list. That is, find those elements that do not have duplicates in the list. For example, in the list $[3, 6, 9, 2, 3, 9, 1, 15, 21, 3, 1]$, the unique elements are $[6, 2, 15, 21]$. The order of elements in the output list should be the same as that in the original list.

   (b) What is the time complexity of your algorithm? You may ignore the improvements introduced by language specific implementations (say, using *set* in Python).

2. (a) Given an integer `n` as input, write a program to print the sum of the following series up to n terms.

$$1 + (1 + 2) + (1 + 2 + 3) + \ldots + (1 + 2 + 3 + \ldots + n)$$

   (b) What is the time complexity of your code?

3. (a) Write a program to print all the most frequently occurring characters in a given string, as a list.
   For example, if the input string is "example",the output should be [e]. If the input string is "exist", then the output should be [e,x,i,s,t].

   (b) What is the complexity of your code?

Ex no: 2
12-03-2023

Name: M.Rohith
3122 21 5001 085

SOLUTIONS

1. a) Program code:

```
l1=[]
n=int(input("Enter the no. of elements in the list: "))
for i in range(n):
    l1.append(int(input("Enter the element: ")))
print("The original list is: ",l1)
l2=[]
for i in range(n):
    if l1.count(l1[i])>1:
        continue
    else:
        l2.append(l1[i])
print("The unique elements are: ",l2)
```

Output:

```
PS C:\Rohith\Backup\Desktop\SEM 4\Design and Analysis of Algorithms lab\Assignment-2> python 1a.py
Enter the no. of elements in the list: 11
Enter the element: 3
Enter the element: 6
Enter the element: 9
Enter the element: 2
Enter the element: 3
Enter the element: 9
Enter the element: 1
Enter the element: 15
Enter the element: 21
Enter the element: 3
Enter the element: 1
The original list is:  [3, 6, 9, 2, 3, 9, 1, 15, 21, 3, 1]
The unique elements are:  [6, 2, 15, 21]
```

b) Time complexity:

The time complexity of the given Python code is O(n^2).

The time complexity of append method is O(1) and that of count method is O(n).

Name: M.Rohith
3122 21 5001 085

The first loop that runs from 0 to n-1 takes O(n) time. Inside this loop, the append method is called n times, which also takes O(n) time in total.

The second loop also runs from 0 to n-1, and for each iteration, the count method is called on the list, which takes O(n) time in the worst case. Therefore, the overall time complexity of the second loop is O(n^2).

Since the first loop and the second loop both take O(n^2) time, the overall time complexity of the code is O(n^2).

2. a) Program code:

```
n=int(input("Enter the value of n: "))
sum=0
for i in range(1,n+1):
    for j in range(1,i+1):
        sum+=j
        print(j,"+",end=" ")
print("\nThe sum is: ",sum)
```

Output:

```
PS C:\Rohith\Backup\Desktop\SEM 4\Design and Analysis of Algorithms lab\Assignment-2> python 2a.py
Enter the value of n: 3
1 + 1 + 2 + 1 + 2 + 3 +
The sum is:  10
```

b) Time complexity:

The time complexity of the given Python code is O(n^2).

The outer loop runs n times, so it has a time complexity of O(n).

The inner loop runs i times for each iteration of the outer loop. Therefore, the time complexity of the inner loop is:

1 + 2 + 3 + ... + n

Name: M.Rohith
3122 21 5001 085

This is a triangular number, which can be expressed as n(n+1)/2. So, the time complexity of the inner loop is O(i), which is equal to O(n) in the worst case when i=n.

Inside the inner loop, a constant amount of work is performed, which takes O(1) time.

Therefore, the overall time complexity of the code is O(n^2).

3. a) Program code:

```
str=input("Enter the string: ")
l1=list(str)
l2=[]
for i in range(len(l1)):
    l2.append(l1.count(l1[i]))
max=max(l2)
l3=[]
for i in range(len(l2)):
    if (max==l2[i] and l1[i] not in l3):
        l3.append(l1[i])
print("The list is: ",l3)
```

Output:

```
PS C:\Rohith\Backup\Desktop\SEM 4\Design and Analysis of Algorithms lab\Assignment-2> python 3a.py
Enter the string: example
The list is:  ['e']
PS C:\Rohith\Backup\Desktop\SEM 4\Design and Analysis of Algorithms lab\Assignment-2> python 3a.py
Enter the string: exist
The list is:  ['e', 'x', 'i', 's', 't']
```

b) Time complexity:

The time complexity of the given Python code is O(n^2), where n is the length of the input string.

The first loop runs n times, and the append method is called on the list, which takes O(1) time in each iteration. Therefore, the time complexity of the first loop is O(n).

The second loop also runs n times. Inside this loop, the count method is called on the list, which takes O(n) time in the worst case when the element is not found in the list. Therefore, the time complexity of the second loop is O(n^2).

The max function is called once, which iterates through the list of counts to find the maximum count. This takes O(n) time.

The third loop also runs n times. Inside this loop, the append method is called on the list, which takes O(1) time in each iteration. The not in operation also takes O(n) time in the worst case when the element is not found in the list. Therefore, the time complexity of the third loop is O(n^2).

Therefore, the overall time complexity of the code is O(n^2).