Page No.:
Date: / /

Lecture 20

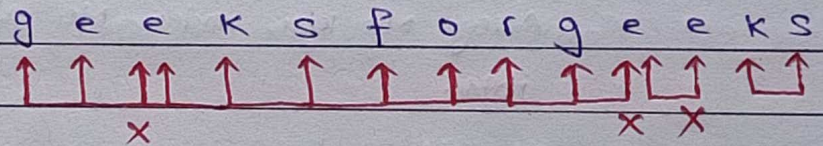
String : Hard problem

* Length of the longest substring

Given a string,
Str = geeksforgeeks

Task : Find largest substring which have all elements are distinct.

geeksgeeks



length = 7

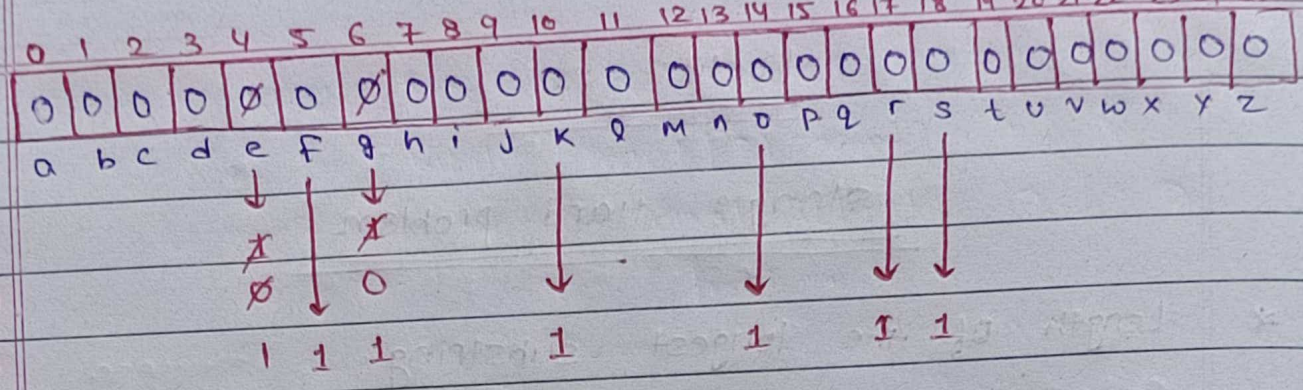
ge, eksforg, eks

↓

Max = 7

g e e k s f o r g e e k s
↑ ↑
i j

Take 2 pointer
First at index 0 → i
Second at index 1 → j



g e e k s f o r g e e k s

i j j
X

count = 2

Increase j

↳ Jab tak koi element dubara nahi mil jata. or

Array Me kisi index Me 1 Nhi mil jata.

Agar Koi element dubara Mil Jata hai to i ko Aage badhao Jabtak Wo repeated element ko Ek baar cross Na kar Jaye.

Array Me zero karte Jao.

g e e k s f o r g e e k s

i j

Again increase j and same process ko repeat karo.

g e e k s f o r g e e k s

i j j repeat

Count = 7

g e e k s f o r g e e k s

i i Count = 7 j

g e e k s f o r g e e k s

i Count = 7 j

Max = 7

This is continue until
j reach the last element

While (j < str.size())

Code:

```
int count[26];
for (int i = 0; i < 26; i++){
        count[i] = 0;
    }
int first = 0;
int second = 1;
count[s[0] - 'a'] = 1;
int total = 1;
```

```
While (second < s.size()){
    While (count[s[second] - 'a']){
        count[s[first] - 'a'] = 0;
        first++;
    }
    count[s[second] - 'a'] = 1;
    total = max(total, second - first + 1);
    second++;
}
return total;
}
```

T. C : O(N)

* Longest Common prefix in an Array :

N = 4

arr[] = {geeksforgeeks, geeks, geek, geezer}

In this problem, we have to return substring for first at which the substring are same for all the string.

arr[0] = geeksforgeeks
arr[1] = geeks
arr[2] = geek
arr[3] = geezer

arr[0][2] = e

The largest possible prefix is the string which size is minimum.

So, we found first the string which size is minimum.

```
int mn = INT_MAX;
for (int i = 0; i < N; i++) {
        if (mn > arr[i].size()){
            mn = arr[i].size();
    }
}
```

By this we can find the size of min-m string and we check the prefix for only that length.

Now we check all the string and compare for index 0 to mn.

mn = 4

arr[0] = geeksforgeeks
arr[1] = geeks
arr[2] = geek
arr[3] = geezer

↓

We can check in this only because the size of smaller string is 4.

Page No.:
Date: / /

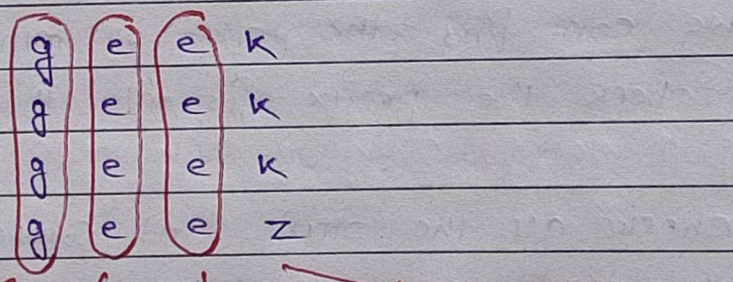
Now, we check for index 0 that all element are same.

If the element in same then we do nothing only we increase the count by 1.

If the element is not same in all string then:

If count is greater than 0 then we return substring from 0 to count - 1.

If count is not greater than 0 then return '-1'.



Same then Count++ Count = 1

Same Count++ Count = 2

Same Count++ Count = 3

Not same
↳ then check for Count

Is Count > 0
↓
Yes
return substr(0, count).

Output: gee

Teacher's Signature

Code :

```
int Count = 0;
int min = INT_MAX;
for (int i = 0; i < N; i++) {
        if (min > arr[i].size()) {
            min = arr[i].size();
        }
}
for (int i= 0; i < min; i++) {
    for (int j=1; j< N; j++) {
        if (arr[j-1][i] != arr[j][i]) {
            if (Count > 0)
                return arr[0].substr(0, Count);
            else
                return "-1";
        }
    }
    Count++;
}
if (Count > 0)
    return arr[0].substr(0, Count);
else
    return "-1";
```

* Longest prefix suffix :

↳ Not be equal to entire string

1) S = "abab"

output = 2

ab → suffix and prefix

2) S = "aaaa"

output = 3

aaa → suffix and prefix

We have return a string which is longest prefix and suffix both for string.

a b c d e

prefix : a, ab, abc, abcd

suffit : e, de, cde, bcde

Approach 1 :

a a a c h a a a

↑ ↑

same, size = 1

a a a c h a a a

↑ ↑

same, size = 2

Teacher's Signature.....

* Approach 2:

a a a c h a a a

↑ ↑

i j

→ a b c a b

↑ ↑ j++;

i j

Not equal

a b c a b

↑ ↑ j++;

i j Not equal.

a b c a b

↑ ↑ i++, j++

i j equal size = 1

a b c a b

↑ ↑ equal

i j size = 2.

We also check j must be the last index then we return the size.

If j is not last then, i become to first.

Teacher's Signature.....

We can check j from every letter which is equal to the first character of the string.

Note:

We can start j only from those index where element present at j index is equal to first character of string.

→ i is always start from first index.

a b c a b d e a b c a b c a b d

Size = 6

return 6

Teacher's Signature.....