

Largest substring where all characters appear at least K times | Set 2

Difficulty Level : Hard • Last Updated : 26 Sep, 2021



Given a string **str** and an integer **K**, the task is to find the length of the longest sub-string **S** such that every character in **S** appears at least **K** times.

Examples:

Input: *str = "aabbba", K = 3*

Output: *6*

Explanation:

In substring aabbba, each character repeats at least k times and its length is 6.

Input: *str = "ababacb", K = 3*

Output: *0*

Explanation:

There is no substring where each character repeats at least k times.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Recommended: Please try your approach on **{IDE}** first, before moving on to the solution.

Naive Approach: We have discussed the Naive Approach in the [previous post](#).

Approach: In this post, we will discuss the approach using [Divide and Conquer](#) technique and [Recursion](#). Below are the steps:

1. Store the [frequency of each characters of the given string](#) in a frequency array of size **26**.
2. Initialize two variables *start* with 0 and *end* which is the length of the string **str**.
3. Iterate over the string from **start** to **end** and count the number of times each character repeats and store it in an array.
4. If any character repeats **less than K times**, then Divide the string into two halves. If *i* is the index of the string where we found that the **string[i]** repeats less than **K times**, then we divide the string into two halves from **start to i** and **i + 1 to end**.
5. Recursively call for the two halves in the above steps i.e., from **start to i** and **i + 1 to end** separately and repeat the **Step 2 and 3** and return the maximum of the two values returned by the above recursive call.
6. If all the characters between **start** and **end** is repeated at least **K** times, then the answer is **end - start**.

Below is the implementation of above approach:

C++

```
// C++ program for the above approach
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
        string s, int k)
{
    int left, right;

    // Array for counting the number of
    // times each character repeats
    // count the number of times each
    // character repeats from start to end
    int count[26] = { 0 };

    // Store the frequency from s[start...end]
    for (int i = start; i < end; i++) {
        count[s[i] - 'a'] += 1;
    }

    // Iterate from [start, end]
    for (int i = start; i < end; i++) {

        if (count[s[i] - 'a'] < k) {

            // Recursive call for left subpart
            left = longestSubstring(start,
                                   i,
                                   s,
                                   k);

            // Recursive call for right subpart
            right = longestSubstring(i + 1,
                                    end,
                                    s,
                                    k);

            // Return maximum of left & right
            return max(left, right);
        }
    }

    // If all the characters are repeated
    // at least k times
    return end - start;
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
// Given String str
string str = "aabbba";
int k = 3;

// Function Call
cout << longestSubstring(0, str.length(),
                        str, k)
    << endl;
return 0;
}
```

Java

```
// Java program for the above approach
import java.util.*;

class GFG{

// Function to find the longest subString
static int longestSubString(int start, int end,
                          String s, int k)
{
    int left, right;

    // Array for counting the number of
    // times each character repeats
    // count the number of times each
    // character repeats from start to end
    int count[] = new int[26];

    // Store the frequency from s[start...end]
    for(int i = start; i < end; i++)
    {
        count[s.charAt(i) - 'a'] += 1;
    }

    // Iterate from [start, end]
    for(int i = start; i < end; i++)
    {
        if (count[s.charAt(i)] < k)
            return -1;
    }
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
        s, k);

    // Recursive call for right subpart
    right = longestSubString(i + 1, end,
                             s, k);

    // Return maximum of left & right
    return Math.max(left, right);
}

// If all the characters are repeated
// at least k times
return end - start;
}

// Driver Code
public static void main(String[] args)
{
    // Given String str
    String str = "aabbba";
    int k = 3;

    // Function Call
    System.out.print(longestSubString(0, str.length(),
                                       str, k) + "\n");
}

// This code is contributed by Amit Katiyar
```

Python3

```
# Python3 program for the above approach

# Function to find the longest substring
def longestSubString(start, end, s, k):
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
count = [0 for i in range(26)]

# Store the frequency from s[start...end]
for i in range(start, end):
    count[ord(s[i]) - ord('a')] += 1

# Iterate from [start, end]
for i in range(start, end):
    if(count[ ord(s[i]) - ord('a')] < k):

        # Recursive call for left subpart
        left = longestSubString(start, i,
                                s, k)

        # Recursive call for right subpart
        right = longestSubString(i + 1, end,
                                s, k)

        # Return maximum of left & right
        return max(left, right)

# If all the characters are repeated
# at least k times
return end - start

# Driver Code

# Given String str
str = "aabbba"
k = 3

# Function call
print(longestSubString(0, len(str), str, k))

# This code is contributed by dadimadhav
```

C#

```
// C# program for the above approach
using System;
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
static int longestSubString(int start, int end,
                           string s, int k)
{
    int left, right;

    // Array for counting the number of
    // times each character repeats
    // count the number of times each
    // character repeats from start to end
    int []count = new int[26];

    // Store the frequency from s[start...end]
    for(int i = start; i < end; i++)
    {
        count[s[i] - 'a'] += 1;
    }

    // Iterate from [start, end]
    for(int i = start; i < end; i++)
    {
        if (count[s[i] - 'a'] < k)
        {

            // Recursive call for left subpart
            left = longestSubString(start, i,
                                   s, k);

            // Recursive call for right subpart
            right = longestSubString(i + 1, end,
                                    s, k);

            // Return maximum of left & right
            return Math.Max(left, right);
        }
    }

    // If all the characters are repeated
    // at least k times
    return end - start;
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
// Given String str
string str = "aabbba";
int k = 3;

// Function call
Console.WriteLine(longestSubString(0, str.Length,
                                   str, k) + "\n");
}
}

// This code is contributed by rutvik_56
```

Javascript

```
<script>

// JavaScript program for the above approach
// Function to find the longest subString
function longestSubString(start, end, s, k)
{
    var left, right;

    // Array for counting the number of
    // times each character repeats
    // count the number of times each
    // character repeats from start to end
    var count = new Array(26);

    // Store the frequency from s[start...end]
    for(var i = start; i < end; i++)
    {
        count[s.charAt(i) - 'a'] += 1;
    }

    // Iterate from [start, end]
    for(var i = start; i < end; i++)
    {
        if (count[s.charAt(i) - 'a'] < k)
        {
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !


```
// Recursive call for right subpart
right = longestSubString(i + 1, end,
                        s, k);

// Return maximum of left & right
return Math.max(left, right);
}
}

// If all the characters are repeated
// at least k times
return end - start;
}

// Driver Code
// Given String str
var str = "aabbba";
var k = 3;

// Function Call
document.write(longestSubString(0, str.length, str, k) + "\n");

// this code is contributed by shivanisinghss2110

</script>
```

Output:

6

Time Complexity: $O(N \cdot \log_2 N)$

**DSA Self-Paced
Course**

- ✓ Curated by experts
- ✓ Trusted by 1 Lac+ students.

Enrol Now



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Like 6

Next

**Largest sub-string where all
the characters appear at
least K times**

RECOMMENDED ARTICLES

Page : 1 2 3

- 01

Longest substring where all the characters appear at least K times | Set 3
10, Feb 21
- 02

Largest sub-string where all the characters appear at least K times
07, Jun 19
- 03

Check if substring S1 appear after any occurrence of substring S2 in given sentence
05, Jan 22
- 04

Count pairs (p, q) such that p occurs in array at least q times and q occurs at least p times
25, Feb 19
- 05

Remove characters that appear more than k times
07, Nov 18
- 06

Given an array of size n and a number k, find all elements that appear more than n/k times
31, May 13
- 07

Remove elements that appear strictly less than k times
10, Nov 18
- 08

Count possible N-digit numbers such that each digit does not appear more than given number of times consecutively
13, Jan 21

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Article Contributed By :

 <https://media.geeksforgeeks.org/auth/avatar.png>
@ruthvicksuresh125

Vote for difficulty

Current difficulty : [Hard](#)

Easy

Normal

Medium

Hard

Expert

Improved By : [amit143katiyar](#), [dadimadhav](#), [rutvik_56](#), [as5853535](#),
[shivanisinghss2110](#), [ruhelaa48](#)

Article Tags : [frequency-counting](#), [Hash](#), [strings](#), [substring](#), [Algorithms](#),
[Analysis](#), [Arrays](#), [Divide and Conquer](#), [Hash](#), [Recursion](#),
[Searching](#), [Strings](#)

Practice Tags : [Arrays](#), [Searching](#), [Hash](#), [Strings](#), [Recursion](#),
[Divide and Conquer](#), [Algorithms](#)

Improve Article

Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

[About Us](#)
[Careers](#)
[In Media](#)
[Contact Us](#)
[Privacy Policy](#)
[Copyright Policy](#)

News

[Top News](#)
[Technology](#)
[Work & Career](#)
[Business](#)
[Finance](#)
[Lifestyle](#)
[Knowledge](#)

Web Development

[Web Tutorials](#)

Learn

[Algorithms](#)
[Data Structures](#)
[SDE Cheat Sheet](#)
[Machine learning](#)
[CS Subjects](#)
[Video Tutorials](#)
[Courses](#)

Languages

[Python](#)
[Java](#)
[CPP](#)
[Golang](#)
[C#](#)
[SQL](#)
[Kotlin](#)

Contribute

[Write an Article](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

NodeJS

@geeksforgeeks , Some rights reserved

Do Not Sell My Personal Information

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !