Largest lexicographical string with at most K consecutive elements

Difficulty Level : Hard  •  Last Updated : 04 Jun, 2021

Given a string S, the task is to find the largest lexicographical string with no more than K consecutive occurrence of an element by either re-arranging or deleting the elements.

Examples:

Input: S = “baccc”  
K = 2  
Output: Result = “ccbcba”  
Explanation: Since K=2, a maximum of 2 same characters can be placed consecutively.  
No. of c’ = 3.  
No. of b’ = 1.  
No. of a’ = 1.  
Since the largest lexicographical string has to be printed, therefore, the answer is “ccbc a”.

Input: S = “xxxxxaz”  
K = 3  
Output: result = “zxxxxa”

Recommended: Please try your approach on (IDE) first, before moving on to the solution.

Approach:
1. Form a frequency array of size 26, where index i is chosen using (a character in a string – ‘a’).
2. Initialize an empty string to store corresponding changes.
3. For i=25 to 0, do:
   • If frequency at index i is greater than k, then append (i + ‘a’) K-times. Decrease frequency by K at index i find the next greatest priority element and append to answer and decrease the frequency at the respective index by 1.
   • If frequency at index i is greater than 0 but less than k, then append (i + ‘a’) times its frequency.
   • If frequency at index i is 0, then that index cannot be used to form an element and therefore check for the next possible highest priority element.
ord('a') == 1

# Empty string of
# string class type
ans = ''

# Loop to iterate over
# maximum priority first.
l = 25
while l >> 0:
    # If frequency is greater than
    # or equal to k.
    if (frequency_array[i] > k):
        # Temporary variable to
        # operate in-place of k.
        temp = k
        st = chr(l + ord('a'))
        while (temp > 0):
            # concatenating with the
            # resultant string ans.
            ans += st
            temp -= 1
        frequency_array[i] -= k
        # Handling k case by adjusting
        # with just smaller priority
        # element.
        j = i - 1
        while (frequency_array[j] <= 0 or
               j == 0)
            j -= 1
        # Condition to verify if index
        # j does have frequency
        # greater than 0;
        if (frequency_array[j] > 0 and
            j >= 0):
            strl = chr(j + ord('a'))
            ans += strl
            frequency_array[j] -= 1
        else:
            # if no such element is found
            # than string can not be
            # processed further.
            break
    # If frequency is greater than 0
    # and less than k.
    elif (frequency_array[i] > 0):
        # Here we don’t need to fix k
        # consecutive element criteria.
        temp = frequency_array[i]
        frequency_array[i] = temp
        st = chr(l + ord('a'))
        while (temp > 0):
            ans += st
            temp -= 1
    # Otherwise check for next
    # possible element.
    else:
        i = l

return ans

# Driver code
if __name__ == "__main__":
    S = "AAABBBCC"
    k = 3
    print (getLargestString(S, k))

# This code is contributed by Chitranshal

Output

```
AAABBBCC
```

**Time Complexity:** O(N)

Attention reader! Don’t stop learning now. Get hold of all the important DSA concepts with the **DSA Self Paced Course** at a student-friendly price and become industry ready. To complete your preparation from learning a language to DS Algo and many more, please refer **Complete Interview Preparation Course**.
Count of triplets \((a, b, c)\) in the Array such that \(a\) divides \(b\) and \(b\) divides \(c\)

Smallest number to make Array sum at most \(K\) by dividing each element

RECOMMENDED ARTICLES

01. Lexicographical smallest number after at most \(K\) consecutive swaps
   15, Jul 20

02. Generate a Binary String without any consecutive 0's and at most \(K\) consecutive 1's
   29, Jul 20

03. Lexicographically largest string possible consisting of at most \(K\) consecutive similar characters
   18, Nov 20

04. Largest lexicographic array with at most \(K\) consecutive swaps
   02, Apr 18

05. Lexicographical concatenation of all substrings of a string
   18, Jan 18

06. Find the \(k\)-th string in lexicographical order consisting of \(n\)-2 X's and 2 Y's
   10, May 20

07. Print all lexicographical greater permutations of a given string
   22, Jun 20

08. K-th lexicographical string of given length
   26, May 20

Article Contributed By:

@kunal_76

Vote for difficulty
Current difficulty: Best

Easy Normal Medium Hard Expert

Improved By: shikhosingrope, princisinh, ukasp, avanitrachhadiya2155

Article Tags: lexicographic-ordering, Algorithms, Arrays, Greedy, Hash, Strings

Practice Tags: Arrays, Hash, Strings, Greedy, Algorithms

Improve Article Report Issue