

Nagarro – 5 September

SD 1 -

mettl Nagarro Bootcamp

Section 1 of 2 SD ▾

You are given a string **S** consisting of lower-case letters and an integer **K**.

Your task is to maximize and return the length **L**, where **L** is the length of a substring such that all substrings of String **S** with length **L** have at most **K** distinct characters.

Note: A substring is a contiguous sequence of characters within a string.

Input Specification:

- input1:** An integer value denoting the length of string **S**.
- input2:** An integer value **K**.
- input3:** The string **S**.

Output Specification:

Return the maximum value of length **L**.

Example 1:

- input1:** 9
- input2:** 2
- input3:** baaabbcca

mettl Vanshikaa | Support

Type here to search

1.

SD - 2

Section 1 of 2 SD

Question # 2 [Revisit](#)

How to attempt?
Question :

Moving Apples

There are N number of baskets, where the i^{th} basket contains $\text{input2}[i]$ apples.

You want to move apples between baskets so that all baskets have the same number of apples. Return the minimum number of apples that must be moved.

Note: It is guaranteed that there exists a way to move apples so as to have an equal number of apples in each basket.

Input Specification:

input1: N, an integer value denoting the Number of Baskets
input2: An integer array representing the number of apples in the i^{th} basket

Output Specification:

Return the minimum number of apples that must be moved so that all baskets have the same number of apples

Example 1:

input1: 2
input2: {1,3}

Language

```
1 #i
2 #i
3 #i
4
5 us
6 //
7
8 in
9 {
10
11
12
13 }
14
15
```

Code

Console Ou

Nagarro_Bootcamp coding_05th September 2

meitl

Section 1 of 2 SD

Output Specification:

Return the minimum number of apples that must be moved so that all baskets have the same number of apples

Example 1:

input1: 2
input2: {1,3}

Output: 1

Explanation:

There are 2 baskets with first containing 1 apple and second containing 3 apples. If we shift one apple from the second container to the first container then both will have 2 apples. So only one apple was required to move.

Hence the output returned is 1.

Example 2:

input1: 5
input2: {2849,1620,705,1,30}

Output: 2387

Language

```
1 #inc
2 #inc
3 #inc
4
5 usin
6 // R
7
8 int
9 {
10
11
12
13 }
14
15
```

Code

Console Out

Question # 3 Revisit

How to attempt?
Question :

Minimum Number of Jumps

Given an array of integers, where each element represents the maximum number of jumps that can be taken forward from that element. Find the minimum number of jumps to reach the end of the array (starting from the first element).

If an element is 0, then no jump can be made from that element. If it is not possible to reach the end, then output -1.

Input Specification:

input1: number of elements in array **input2:** (2 <= input1 <= 1000)
input2: an integer array

Output Specification:

Return the minimum number of jumps to reach the end.

Example 1:

input1: 3
input2: {2,1,1}

Output: 1

Return the minimum number of jumps to reach the end.

Example 1:

input1: 3
input2: {2,1,1}

Output: 1

Explanation: The first element is 2, this means that 2 jumps can be taken forward from this element. With this, we reach the last element. Hence, the total number of required jumps is 1.

Example 2:

input1: 9
input2: {1,3,6,1,0,9,8,7,6}

Output: 3

Explanation: The first element is 1, this means that 1 jump can be taken and reach element 3. Now, 3 jumps can be taken forward from this element. On taking the first step, we get element 6. If 6 jumps are taken from here itself, the last element of the array will be reached. Hence, the total number of required jumps will be 3.

