Max Consecutive Ones - 485

Given a binary array nums, return the maximum number of consecutive 1's in the array.

Input: nums = [1,1,0,1,1,1]

Output: 3

Explanation: The first two digits or the last three digits are consecutive 1s. The maximum number of consecutive 1s is 3.

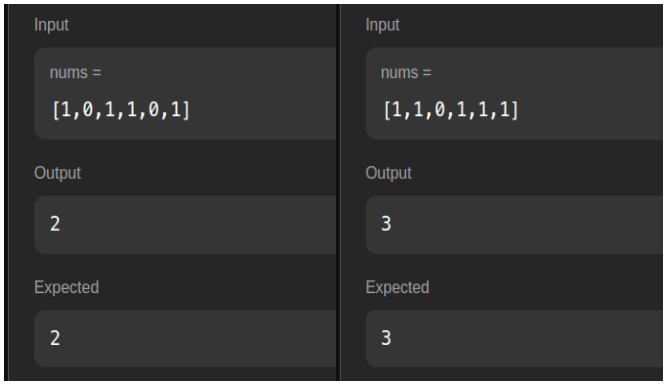
Solution:

```python
class Solution:
    def findMaxConsecutiveOnes(self, nums: List[int]) -> int:
        ans = 0
        l = 0
        for r in range(0, len(nums)):
            if nums[l]==0:
                l+=1
                r = l
            elif nums[r]==0:
                ans = max(ans, r-l)
                l = r
        if nums[-1]:
            ans = max(ans,r-l+1)
        return ans


```

Approach:

Slide through the array until you find the first one to make it 'l' (variable). Then start traverse until you find another 0. Now, calculate the answer and make the current position as l. And again start find a 1 to make it as l. Repeat.

Time Complexity: O(n) Space Complexity: O(1)



Longest Substring with At Least K Repeating Characters - 595

Given a string s and an integer k, return the length of the longest substring of s such that the frequency of each character in this substring is greater than or equal to k. If no such substring exists, return 0.

Input: s = "aaabb", k = 3
Output: 3

Explanation: The longest substring is "aaa", as 'a' is repeated 3 times.

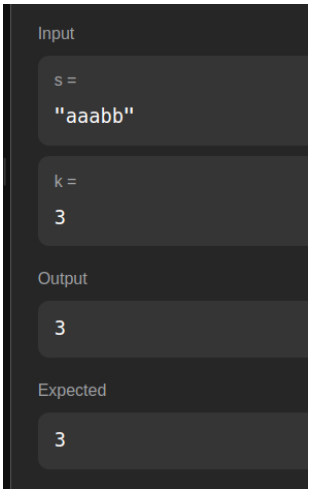
Solution:

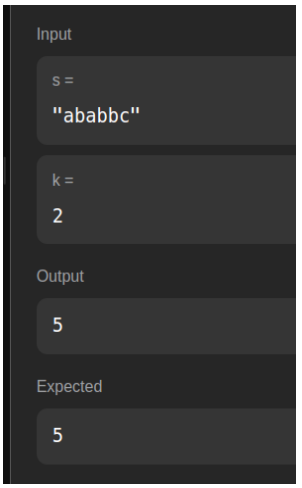
```java
class Solution {
    public int longestSubstring(String s, int k) {
        int n = s.length();
        if(n<k) return 0;

        HashMap<Character, Integer> freq = new HashMap<>();
        for(char ch: s.toCharArray()){
            if(!freq.containsKey(ch)) freq.put(ch, 0);
            freq.put(ch, freq.get(ch)+1);
        }
        for(int i=0;i<n;i++){
            if(freq.get(s.charAt(i))<k){
                int lf = longestSubstring(s.substring(0,i),k);
                int rh = longestSubstring(s.substring(i+1,n),k);
                return Math.max(lf,rh);
            }
        }
        return s.length();
    }
}
```

Approach:

Calculate Frequency. Traverse through the string until you find the invalid character. Once you find it, break the string and repeat for the left and right substring.

Time Complexity: O(N**2) , Space: O(N)





Subarray Sum Equals K - 560

Given an array of integers nums and an integer k, return the total number of subarrays whose sum equals to k. A subarray is a contiguous non-empty sequence of elements within an array.

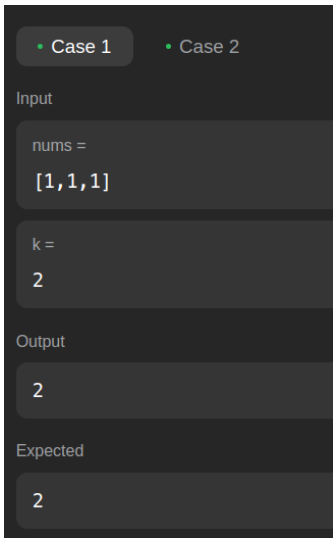
Input: nums = [1,2,3], k = 3

Output: 2

Solution:

```python
class Solution:
    def subarraySum(self, nums: List[int], k: int) -> int:
        ans = 0
        now = 0
        d = {0:1}
        for r in nums:
            now+=r
            if (now-k in d):
                ans+=d[now-k]
            d[now] = d.get(now,0)+ 1
        return ans
```

Output:



Approach:
Iterates through the list, for each cumulative sum, it checks if the difference between the current sum and k is present in the dictionary. If found, it adds the corresponding frequency to the answer count. The dictionary is continuously updated to keep track of cumulative sum frequencies.

Time Complexity: O(N)

Space Complexity: O(N)