**MEDIUM**

**438. Find All Anagrams in a String**

**Input:** s = "cbaebabacd", p = "abc"

**Output:** [0,6]

**Explanation:**

The substring with start index = 0 is "cba", which is an anagram of "abc".

The substring with start index = 6 is "bac", which is an anagram of "abc".

**424. Longest Repeating Character Replacement**

**Input:** s = "ABAB", k = 2

**Output:** 4

**Explanation:** Replace the two 'A's with two 'B's or vice versa

**395. Longest Substring with At Least K Repeating Characters**

**Input:** s = "ababbc", k = 2

**Output:** 5

**Explanation:** The longest substring is "ababb", as 'a' is repeated 2 times and 'b' is repeated 3 times.

**340. Longest Substring with At Most K Distinct Characters**

**Input:** s = " aabacbebebe", k = 3

**Output:** 7

**3. Longest Substring Without Repeating Characters**

**Input:** s = "abcabcbb"

**Output:** 3

**Explanation:** The answer is "abc", with the length of 3.

**1004. Max Consecutive Ones III**

**Input:** nums = [1,1,1,0,0,0,1,1,1,1,0], k = 2

**Output:** 6

**Explanation:** [1,1,1,0,0,**1**,1,1,1,1,**1**]

Bolded numbers were flipped from 0 to 1. The longest subarray is underlined.

**1423. Maximum Points You Can Obtain from Cards**

**Input:** cardPoints = [1,2,3,4,5,6,1], k = 3

**Output:** 12

**Explanation:** After the first step, your score will always be 1. However, choosing the rightmost card first will maximize your total score. The optimal strategy is to take the three cards on the right, giving a final score of 1 + 6 + 5 = 12.

**358. Number of Substrings Containing All Three Characters**

**Input:** s = "abcabc"

**Output:** 10

**Explanation:** The substrings containing at least one occurrence of the characters *a*, *b* and *c are "*abc*", "*abca*", "*abcab*", "*abcabc*", "*bca*", "*bcab*", "*bcabc*", "*cab*", "*cabc*"* and *"*abc*"* (**again**)*.*

**567. Permutation in String**

**Input:** s1 = "ab", s2 = "eidbaooo"

**Output:** true

**Explanation:** s2 contains one permutation of s1 ("ba").

**HARD**

**239. Sliding Window Maximum**

**Input:** nums = [1,3,-1,-3,5,3,6,7], k = 3

**Output:** [3,3,5,5,6,7]

**Explanation:**

Window position Max

--------------- -----

[1 3 -1] -3 5 3 6 7 **3**

1 [3 -1 -3] 5 3 6 7 **3**

1 3 [-1 -3 5] 3 6 7  **5**

1 3 -1 [-3 5 3] 6 7 **5**

1 3 -1 -3 [5 3 6] 7 **6**

1 3 -1 -3 5 [3 6 7] **7**

**727. Minimum Window Subsequence**

**Input:**   
S = "abcdebdde", T = "bde"  
**Output:** "bcde"  
**Explanation:**   
There are many substrings with "bde" but the smallest amongst them is "bcde" and "bdde" of length 4. Out of these 2, "bcde" occurs first, Hence it is the answer.

**76. Minimum Window Substring**

**Input:** s = "ADOBECODEBANC", t = "ABC"

**Output:** "BANC"

**Explanation:** The minimum window substring "BANC" includes 'A', 'B', and 'C' from string t.

**992. Subarrays with K Different Integers**

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

**Output:** 7

**Explanation:** Subarrays formed with exactly 2 different integers: [1,2], [2,1], [1,2], [2,3], [1,2,1], [2,1,2], [1,2,1,2]