sliding window pattern based problem that are mostly asked in interview question

The sliding window pattern is a useful technique for solving problems involving arrays or lists where you need to maintain a subset of the elements and adjust its size or position. Here are some common sliding window problems often asked in interviews:

1. Maximum Sum Subarray of Size K

Problem: Given an array of integers and a number `k`, find the maximum sum of a contiguous subarray of size `k`.

Example:

- Input: `arr = [1, 2, 3, 4, 5]`, `k = 3`
- Output: `12` (since the subarray `[3, 4, 5]` has the maximum sum)

2. Minimum Window Substring

Problem: Given two strings `s` and `t`, find the minimum window in `s` which will contain all the characters of `t`.

Example:

- Input: `s = "ADOBECODEBANC"`, `t = "ABC"`
- Output: `"BANC"`

3. Longest Substring Without Repeating Characters

Problem: Given a string, find the length of the longest substring without repeating characters.

Example:

- Input: `s = "abcabcbb"`
- Output: `3` (The longest substring is `"abc"`)

4. Find All Anagrams in a String

Problem: Given two strings `s` and `p`, find all start indices of `p`'s anagrams in `s`.

Example:

- Input: `s = "cbaebabacd"`, `p = "abc"`
- Output: `[0, 6]` (The anagrams of `"abc"` are `"cba"` and `"bac"`)

5. Longest Substring with At Most K Distinct Characters

Problem: Given a string and an integer `k`, find the length of the longest substring that contains at most `k` distinct characters.

Example:

- Input: `s = "eceba"`, `k = 2`
- Output: `4` (The longest substring is `"eceb"`)

6. Subarrays with Sum K

Problem: Given an array of integers and an integer `k`, find the number of subarrays that sum up to `k`.

Example:

- Input: `arr = [1, 1, 1]`, `k = 2`
- Output: `2` (The subarrays `[1, 1]` are the ones that sum up to `2`)

7. Longest Repeating Character Replacement

Problem: Given a string and an integer `k`, find the length of the longest substring that can be achieved by replacing at most `k` characters.

Example:

- Input: `s = "ABAB"`, `k = 2`
- Output: `4` (The longest substring after replacing at most `2` characters is `"BBBB"`)

These problems vary in complexity but share the common theme of using a sliding window to maintain a subset of elements and make decisions based on that subset.