

Sliding Window

5. Sliding Window:

Problem: The sliding window technique is used to solve problems involving subarrays or substrings in an efficient manner.

Approach: Maintain a window of elements and slide it across the array, updating the result as the window moves.

Scenarios:

1. ****Longest Substring Without Repeating Characters:**** Find the longest substring without repeating characters using a sliding window.
2. ****Maximum Sum Subarray of Size K:**** Find the subarray with the maximum sum of a fixed size using a sliding window.
3. ****Sliding Window Maximum:**** Find the maximum element in each sliding window of size `k`.

Java Code for Maximum Sum Subarray of Size K:

```
public class Solution {  
  
    public int maxSumSubarray(int[] arr, int k) {  
  
        int n = arr.length;  
  
        int maxSum = 0, windowSum = 0;  
  
        for (int i = 0; i < k; i++) {  
  
            windowSum += arr[i];  
  
        }  
  
        maxSum = windowSum;  
  
        for (int i = k; i < n; i++) {  
  
            windowSum += arr[i] - arr[i - k];
```

```
        maxSum = Math.max(maxSum, windowSum);  
    }  
    return maxSum;  
}  
}
```

Explanation: The sliding window technique is used to calculate the sum of subarrays of size `k` in linear time.