

Subarrays

4. Subarrays:

Problem: Subarray problems involve finding subarrays that satisfy certain conditions.

Approach: Use sliding window, prefix sums, or dynamic programming to find subarrays efficiently.

Scenarios:

1. **Maximum Subarray Sum (Kadane's Algorithm):** Find the subarray with the maximum sum using dynamic programming.
2. **Subarrays with a Specific Sum:** Find subarrays that sum to a specific value using prefix sums.
3. **Subarrays with Maximum Product:** Find subarrays with the maximum product using a modified version of Kadane's algorithm.

Java Code for Maximum Subarray Sum:

```
public class Solution {  
    public int maxSubArray(int[] nums) {  
        int maxSum = nums[0], currentSum = nums[0];  
        for (int i = 1; i < nums.length; i++) {  
            currentSum = Math.max(nums[i], currentSum + nums[i]);  
            maxSum = Math.max(maxSum, currentSum);  
        }  
        return maxSum;  
    }  
}
```

Explanation: Kadane's algorithm is used to find the subarray with the maximum sum in linear time.