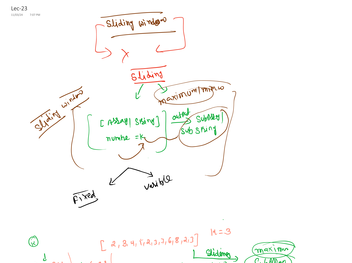
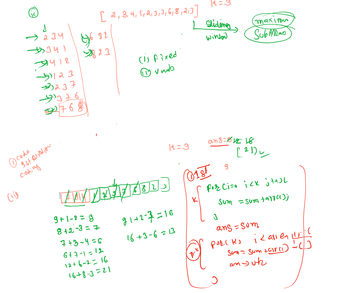
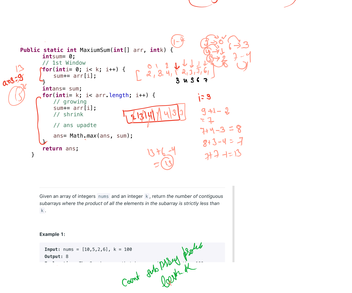
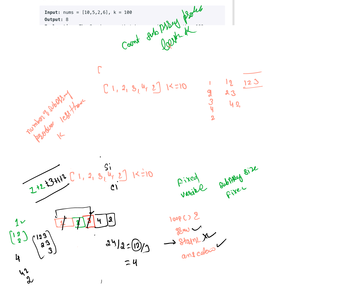
Lec-23








```
Public static int MaxiumSum(int[] arr, int k) {
    int sum = 0;
    // 1st Window
    for(int i= 0; i< k; i++) {
        sum+= arr[i];
    }
    int ans= sum;
    for(int i= k; i< arr.length; i++) {
        // growing
        sum+= arr[i];
        // shrink
        sum-= arr[i-k];
        // ans update
        ans= Math.max(ans, sum);
    }
    return ans;
}
```

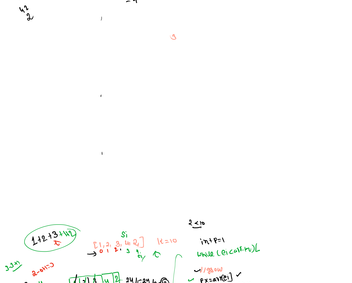


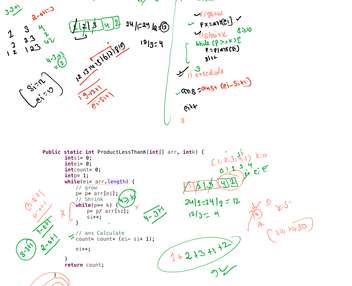
Given an array of integers `nums` and an integer `k`, return the number of contiguous subarrays where the product of all the elements in the subarray is strictly less than `k`.

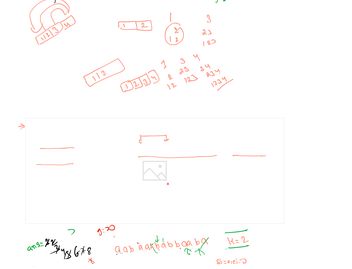
Example 1:

```
Input: nums = [10,5,2,6], k = 100
Output: 8
```

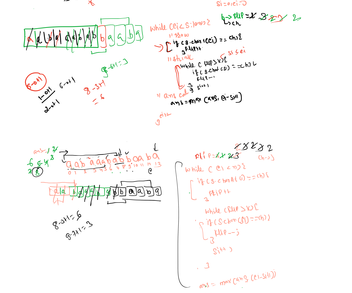
Count sub array product < K

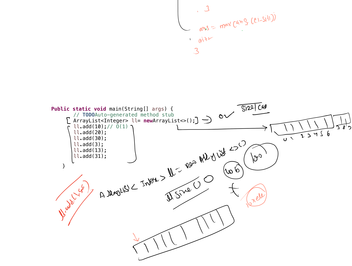


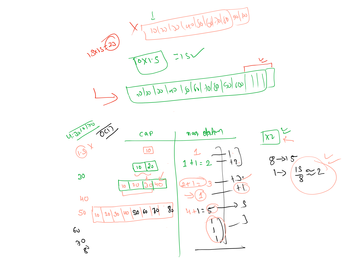





```
Public static int ProductLessThanK(int[] arr, int k) {
    int i= 0;
    int n= 0;
    int count= 0;
    int p= 1;
    while(n< arr.length) {
        // grow
        p= p* arr[n];
        // Shrink
        while(p>= k) {
            p= p/ arr[i];
            i++;
        }
        // ans Calculate
        count+ =count+ (n- i+ 1);
        n++;
    }
    return count;
}
```








```
Public static void main(String[] args) {
    // TODO Auto-generated method stub
    ArrayList<Integer> lm= new ArrayList<>();
    lm.add(20);// 0[1]
    lm.add(20);
    lm.add(30);
    lm.add(33);
    lm.add(33);
    lm.add(33);
}
```