**Sliding window (AUG 29 – sept 8)sept 1st week**

**Level 1: Fixed-Size Window**

👉 These are closest to the one you just solved.

* **LeetCode 643. Maximum Average Subarray I** ✅ (what you did)
* **LeetCode 1343. Number of Sub-arrays of Size K and Average ≥ Threshold**  
  (similar but you check condition instead of max)
* **LeetCode 1456. Maximum Number of Vowels in a Substring of Given Length**  
  (same logic but on strings: count vowels in window of size k)

**🔹 Level 2: Variable-Size Window**

👉 Window can expand or shrink depending on condition.

* **LeetCode 209. Minimum Size Subarray Sum**  
  (find smallest subarray with sum ≥ target)
* **LeetCode 3. Longest Substring Without Repeating Characters**  
  (classic variable window with hash set)
* **LeetCode 1004. Max Consecutive Ones III**  
  (flip at most k zeros → longest subarray of ones)

**🔹 Level 3: Sliding Window + HashMap/Counting**

👉 Add frequency maps for tougher problems.

* **LeetCode 567. Permutation in String**  
  (check if one string’s permutation is in another)
* **LeetCode 438. Find All Anagrams in a String**  
  (find all anagrams = fixed window with hashmap comparison)
* **LeetCode 76. Minimum Window Substring**  
  (very famous — smallest substring containing all chars)

**🔹 How to Analyse Any Sliding Window Problem**

1. **Ask:** Is it about contiguous subarray/substring?
   * Yes → Sliding window candidate.
2. **Check Window Size:**
   * Fixed (like k = 3)? → Easy sliding sum.
   * Variable (depends on sum/condition)? → Expand & shrink with two pointers.
3. **Extra tracking:**
   * Just sum? (easy)
   * Or need frequency/count/map? (harder)