Sliding Window Technique is useful for solving problems in

- array
- string

*Sliding windows are a subset of dynamic programming problems

especially it is considered as a technique that could reduce the time complexity from $O(n^2)$ to $O(n)$

However,

there are two types of sliding Window :

① Fixed window length K

the length of the window is fixed and it asks you to find something in the window

↳ such as the maximum or median number of each window.

Usually we need some kind of variables to maintain the state of the window,

Some are as simple as a integer or it could be as complicated as some advanced data structure such as a list, queue or deque

② Two pointers + criteria:

The window size is not fixed, usually it asks you to find the subarray that meets the criteria

↳ for example, given an array of integers find the number of subarrays whose sum is equal to a target value

How do you identify Sliding Window Problems?

1) The problem will involve a data structure that is ordered and iterable like an array or a string

2) You are looking for some subrange in that array/string like a longest, shortest or target value

8) There is an apparent naive or brute force solution that runs in $O(n^2)$, $O(2^n)$ or some other large complexity

BUT the biggest give away is that the thing you're looking for is often some kind of optimal

↳ like the longest sequence

or shortest sequence of something that satisfies a given condition exactly...

And the amazing thing about sliding window problems is that most of the time they can be solved in $O(N)$ time and $O(1)$ space complexity

Why is this dynamic programming?

- This search for an optimum hints at the relationship between sliding window problems and other dynamic problems.

- You are using the optimal substructure property of the problem to guarantee that an optimal solution to a subproblem can be reused to help you find the optimal solution to a larger problem