

Sliding Window: Fixed

A subset of the Two Pointer Method, but uses left and right pointers to define the bounds of a "window" in iterable data structures like arrays. The window defines the subcomponent, like subarray or substring, and it slides across the data structure in one direction, searching for a subcomponent that meets a certain requirement.



Window indices: [0, 2] with values: [1, 2, 3]

When to use:

- Data Structure: Linear such as Array, Linked List
- Find a Subcomponent of a length

Brute Force:

- Finding all possible subcomponents for an answer using a Nested Loop
 - Outer Loop traverses the array for the first element of the pair
 - Inner Loop traverses the rest of the array to find second element
- Time Complexity is $O(n^2)$ where n is length of the loop (Two Loops)

```
def sliding_window_fixed(nums, window_size):  
    n = len(nums)  
    # Slide the window from the start of the array until the end.  
    for i in range(n - window_size + 1):  
        window = nums[i:i + window_size]  
        print(f"Window indices: [{i}, {i + window_size - 1}] -> Values: {window}")  
  
# Example usage:  
nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
window_size = 3
```

```
sliding_window_fixed(nums, window_size)
```

Real-World Example:

Buffering in Video Streaming

Revision #3

Created 28 December 2024 22:14:24 by victor

Updated 16 March 2025 22:46:39 by victor