



# Sliding Window & Binary Search



Kevin Yu



# Sliding Window



Instead of repeatedly iterating over the same elements, the sliding window maintains a window that moves through the data



window sum =  $4 + 2 + 1 = 7$  (initial calculation)



window sum =  $7 + 7 - 4 = 10$  (update previous sum)



window sum =  $10 + 8 - 2 = 16$  (update previous sum)

# Sliding Window Example - Longest Substring Without Repeating Character

<https://leetcode.com/problems/longest-substring-without-repeating-characters/>

```
input str = "a b c a b f c b b"
```

```
{a}  "a" b c a b f c b b"
```

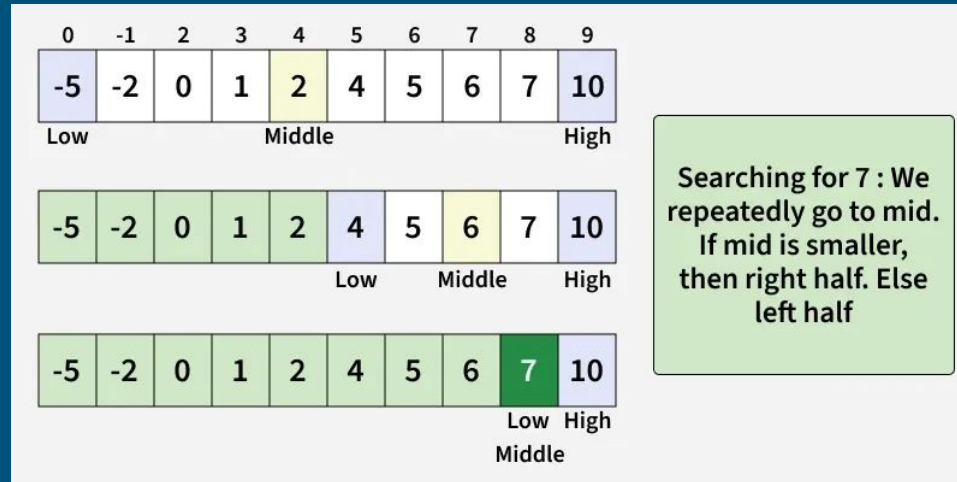


1. Create a empty hashset
2. Keep track of the max window (set) size
3. Create pointer pStart = 0, pEnd = 0
4. While (pEnd != str.size() - 1)
  - a. Check if char at pEnd is in the set
    - if char is not in the set - add it to the set and increment pEnd by 1
    - if char is in the set - remove char at pStart from the set and increment pStart
  - b. if set size > the current max window size, update the max window size
5. return max

# Binary Search

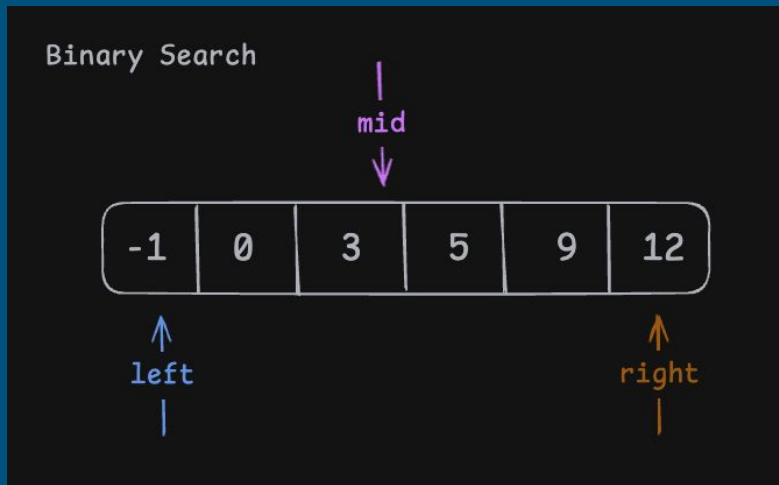


- The data structure must be sorted.
- Time Complexity  $O(\log(n))$



# Binary Search Example - Binary Search

<https://leetcode.com/problems/binary-search/>



# Binary Search Example - Find Minimum in Rotated Sorted Array

<https://leetcode.com/problems/find-minimum-in-rotated-sorted-array>

# Thank You

