@arshgoyalyt

TOP 12 PATTERNS TO ACE ANY CODING INTERVIEW

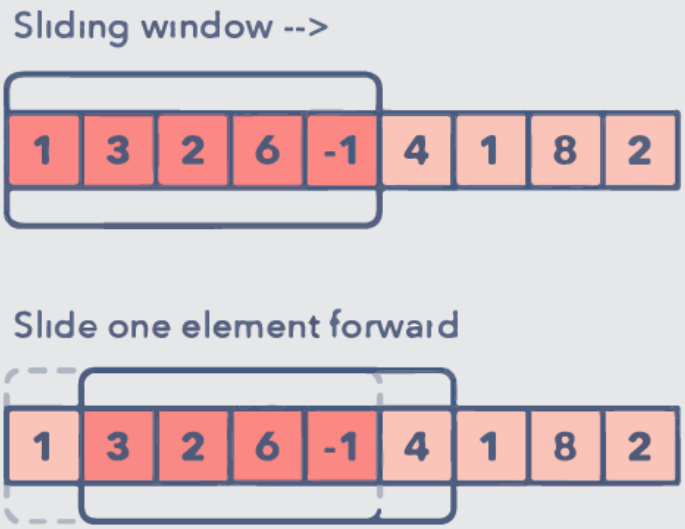
PART - I

SLIDING WINDOW

The **Sliding Window** pattern is used to perform a required operation on a specific window size of a given array or linked list, such as finding the longest subarray containing all 1s.

Sliding Windows start from the 1st element and keep shifting right by one element and adjust the length of the window according to the problem that you are solving.

@arshgoyalyt

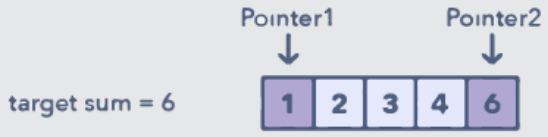


TWO POINTERS OR ITERATORS

Two Pointers is a pattern where two pointers iterate through the data structure in tandem until one or both of the pointers hit a certain condition.

Two Pointers is often useful when searching pairs in a sorted array or linked list; for example, when you have to compare each element of an array to its other elements.

@arshgoyalyt



1 + 6 > target sum, therefore let's decrement Pointer2

1 + 4 < target sum, therefore let's increment Pointer1

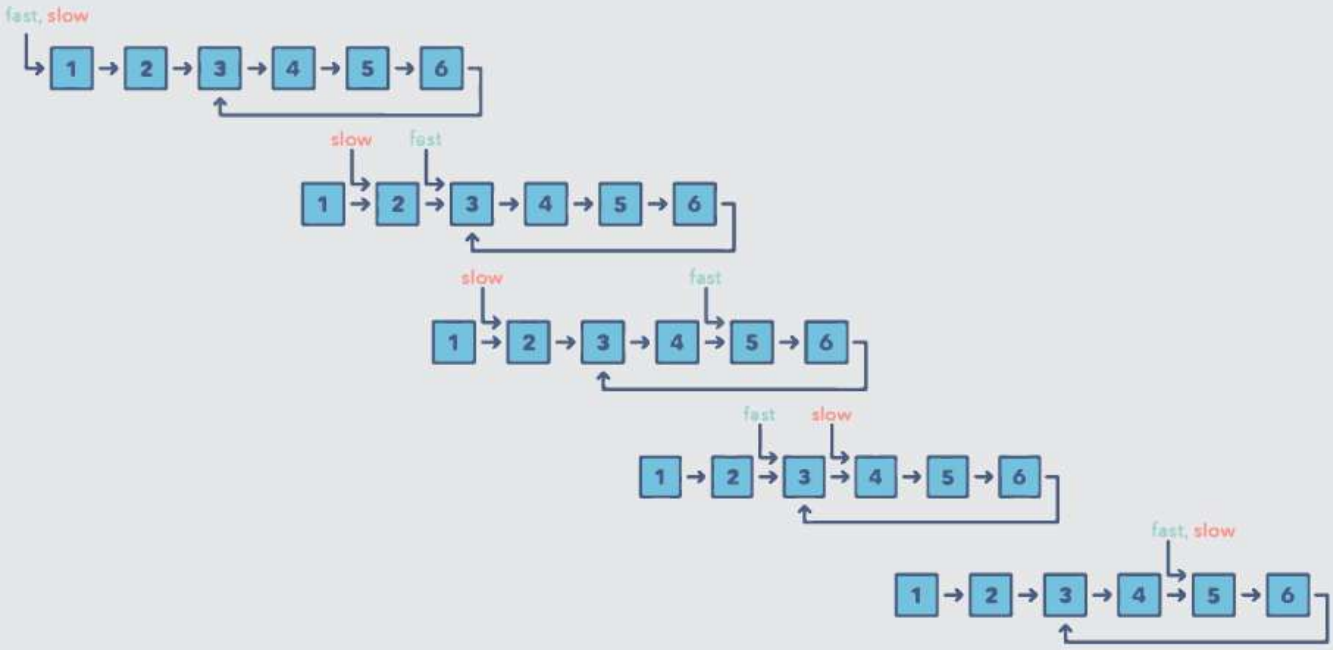
2 + 4 == target sum, we have found our pair

FAST AND SLOW POINTERS

The **Fast and Slow** pointer approach, also known as the **Hare & Tortoise** algorithm, is a pointer algorithm that uses two pointers which move through the array (or sequence/linked list) at different speeds.

This approach is quite useful when dealing with cyclic linked lists or arrays.

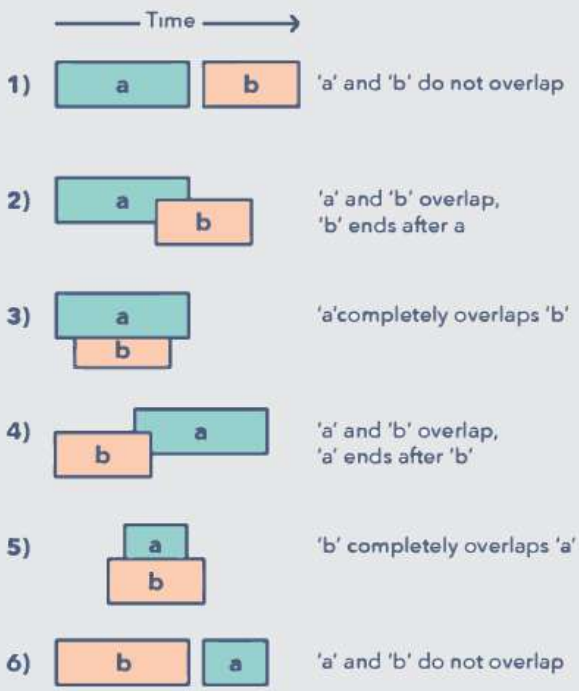
@arshgoyalyt



MERGE INTERVALS

The **Merge Intervals** pattern is an efficient technique to deal with overlapping intervals.
In a lot of problems involving intervals, you either need to find overlapping intervals or merge intervals if they overlap.

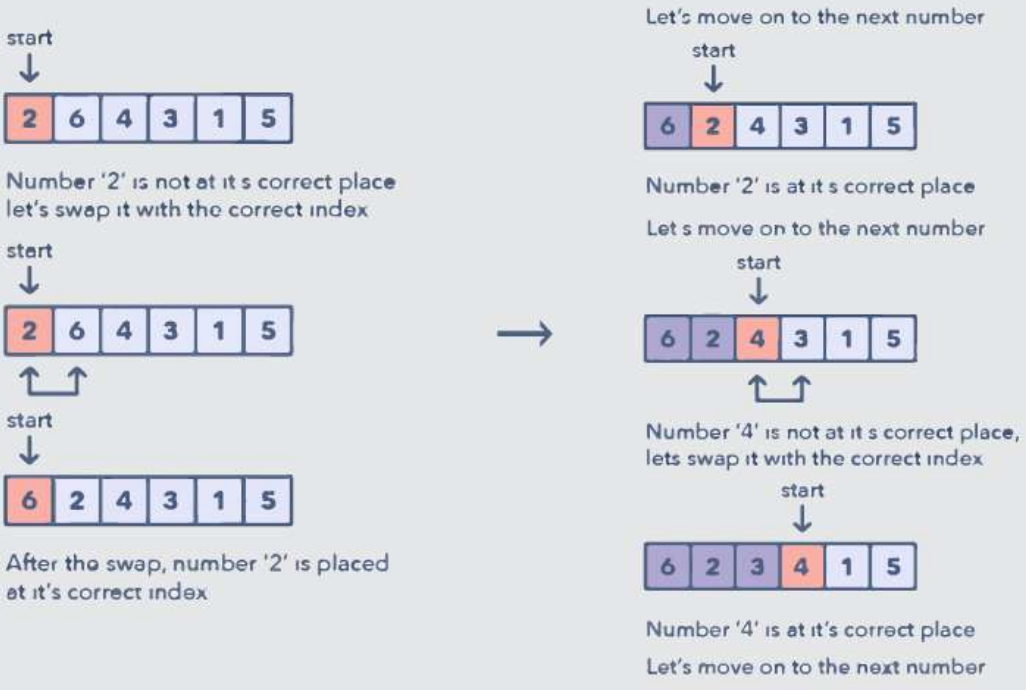
@arshgoyalyt



CYCLIC SORT

The **Cyclic Sort** pattern iterates over the array one number at a time, and if the current number you are iterating is not at the correct index, you swap it with the number at its correct index.

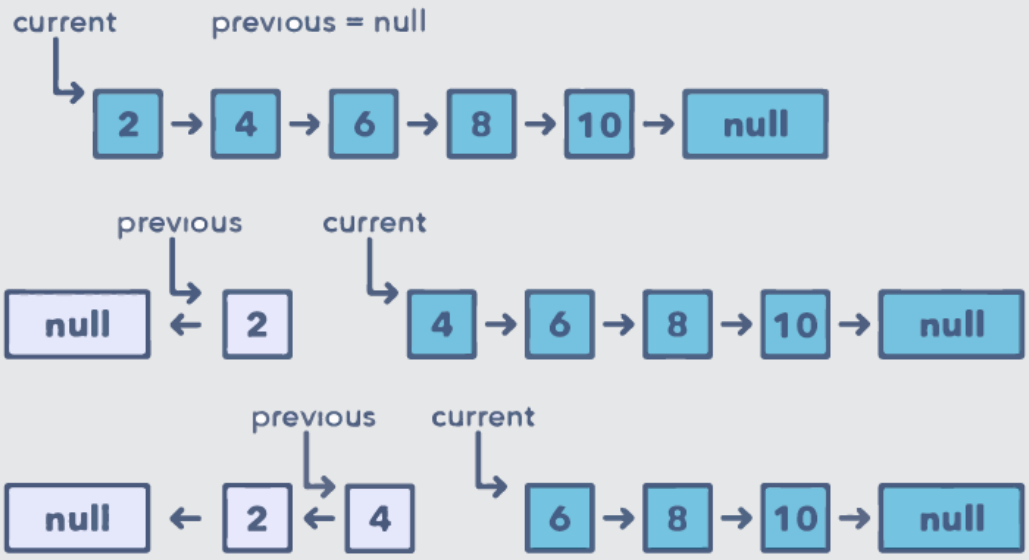
@arshgoyalyt



IN-PLACE REVERSAL OF LINKED LIST

This pattern reverses one node at a time starting with one variable (current) pointing to the head of the linked list, and one variable (previous) will point to the previous node that you have processed.
In a lock-step manner, you will reverse the current node by pointing it to the previous before moving on to the next node.

@arshgoyalyt



Follow this page

Comment down your favourite pattern ?