# Introduction to Two Pointers

↳ Two Pointers pattern is a technique used for solving problems involving sequential data structures (e.g., arrays, linked lists).

↳ It involves maintaining two pointers that traverse the structure in a coordinated way.

↳ The pointers typically start from different positions or move in opposite directions.

↳ They dynamically adjust based on conditions or criteria related to the problem.

↳ This pattern enables efficient exploration of data.

↳ It often leads to optimal time and space complexity solutions.

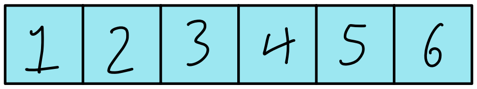
↳ Especially useful when trying to find two elements in an array that satisfy a specific condition.

↳ Should be one of the first strategies considered for such problems.

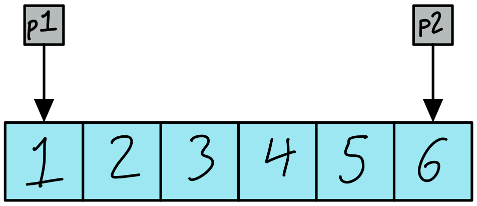
## Examples

↳ Reversing an Array: Given an array of integers, reverse it in place.

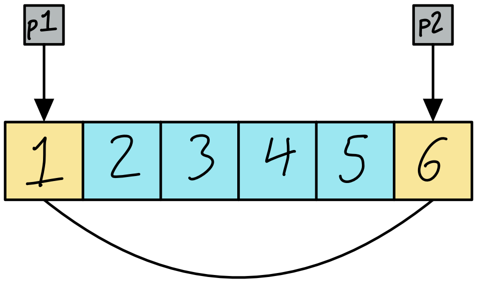
i) The input array



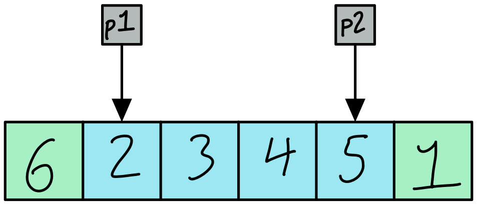
ii) Initialize two pointers, p1 and p2, at the start and end of the array.



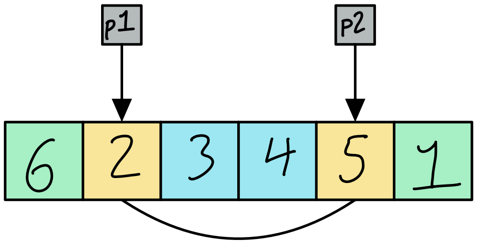
iii) Swap the values of the indexes pointed to by p1 and p2.



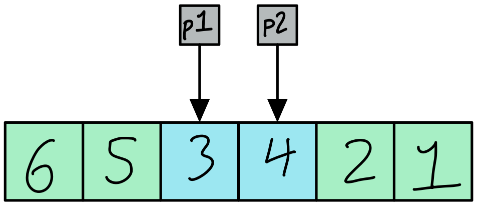
iv) After swapping the values, move the pointers inwards, i.e., increment p1 and decrement p2.



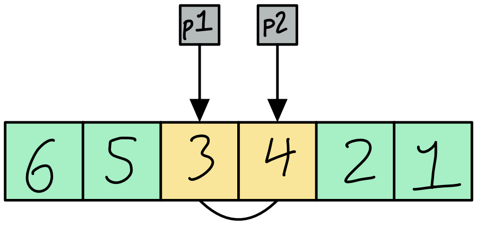
v) Swap the values of the indexes pointed to by p1 and p2.



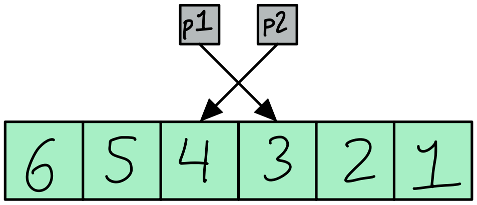
vi) After swapping the values, move the pointers inwards, i.e., increment p1 and decrement p2.



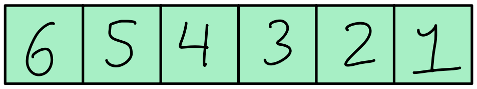
vii) Swap the values of the indexes pointed to by p1 and p2.



viii) After swapping the values, move the pointers inwards, i.e., increment p1 and decrement p2.

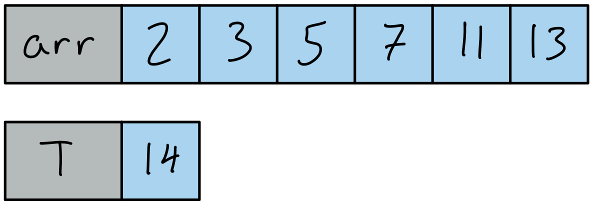


ix) Because the pointers p1 and p2 have surpassed, the algorithm ends. The array has been reversed.

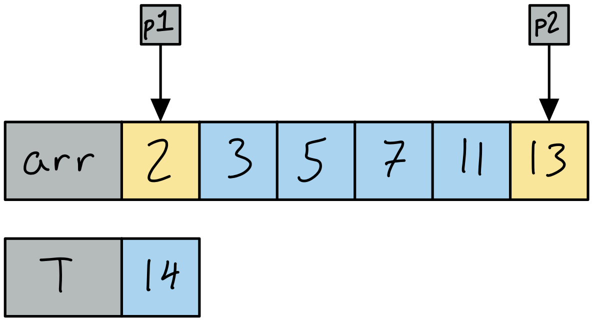


↳ Pair with Given Sum in a Sorted Array: Given a sorted array of integers, find a pair in the array that sums to a number T.

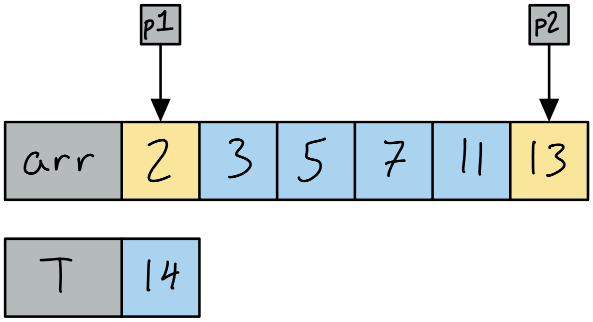
i) Given the following array, we need to find a pair that sums up to 14.



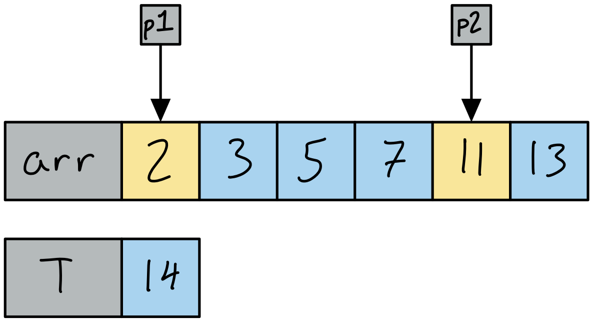
ii) Initialize two pointers, p1 and p2, at the start and end of the array. Calculate the sum of the indexes pointed to by p1 and p2. If the sum is equal to T, we've found the solution pair. If the sum is greater than T, decrement p2. Otherwise, increment p1.



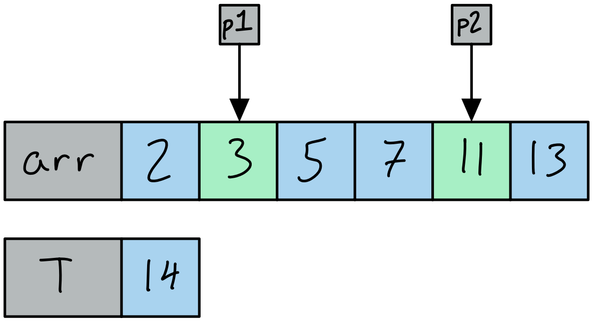
iii) Because the sum of the values pointed to by p1 and p2 is greater than T, decrement p2 to move it inward towards the smaller values in the array.



iv) Now, the sum of the values pointed to by p1 and p2 is less than T, increment p1.



v) The p1 and p2 are pointing to 3 and 11. Since they sum to T, i.e. 14, the solution pair is found.



# Does the Problem Match this Pattern?

↳ Yes, if all the following conditions are true:

i) <u>Linear Data Structure</u>: The input data can be traversed in a linear fashion, such as an array, linked list, or string.

ii) <u>Process Pairs</u>: Process data elements at two different positions simultaneously.

iii) <u>Dynamic Pointer Movement</u>: Both pointers move independently of each other according to certain conditions or criteria. In addition, both pointers might move along the same or two different data structures.