Note 2.

Table of Content:

Topic:

2: Two Pointers or Iterators

Two pointers is a pattern where two iterate through the data structure in tandem (song song) 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.

Two pointers are needed because with just pointer, you would have to continually loop back through the array to find the answer. This back and forth with a single iterator is inefficient for time and space complexity – a concept referred to as asymptotic analysis. While the brute force or naïve solution with 1 pointer would work, it will produce something along the lines of O(n^2). In many cases, 2 pointers can help you find a solution with better space or runtime complexity.

Diagram

Description automatically generated

A picture containing text, clock

Description automatically generated

Diagram

Description automatically generated

--- Squares of a sorted array

I: an integer array

* by nums sorted in non-decreasing order. (aka increasing order)
* there are negative numbers

O: an array of the squares of each number sorted in non-decreasing order

C:

* 1 <= nums.length <= 10^4
* -10^4 <= nums[i] <= 10^4
* Nums is sorted is non-decreasing order

E: empty array.

Intuition

Since the Array A is sorted, loosely speaking it has some negative elements with squares in decreasing order, then some non-negative elements with squares in increasing order.

For example with [-3, -2, -1, 4, 5, 6], we have the negative part [-3, -2, -1] with squares [9, 4, 1], and the positive part [4, 5, 6] with squares [16, 25, 36]. Our strategy is to iterate over the negative part in reverse, and the positive part in the forward direction.

2 pointer technique is different than the sliding window technique are that:

1: 2 pointer technique: 1 pointer starts at the beginning, the other pointer starts at the end

2: sliding window technique is that: 2 pointers start at the same point.

Diagram, rectangle

Description automatically generated

Text, letter

Description automatically generated

Algorithm

We can use 2 pointers to read the positive and negative parts of the array – one pointer j in the positive direction, and another I in the negative direction.

Need to fill the result array.

Now that we are reading 2 increasing arrays (the squares of the elements), we can merge these arrays together using a 2 pointer technique.