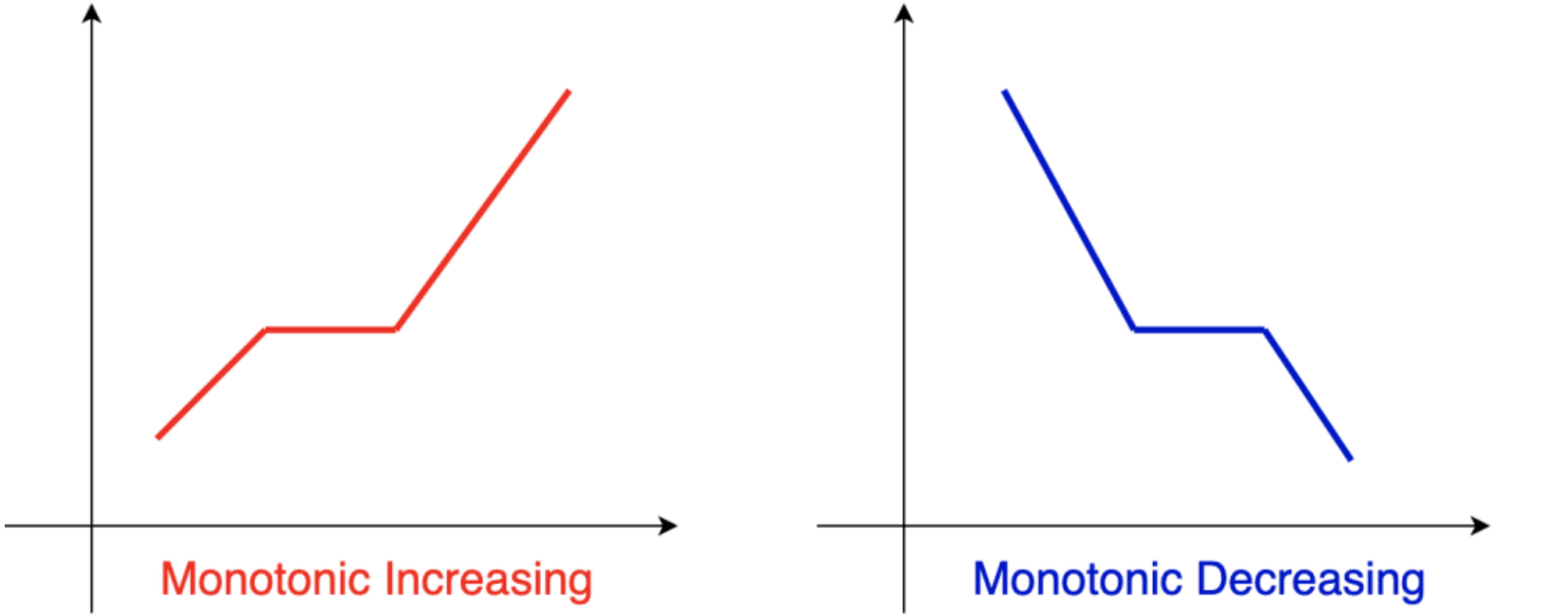

Gopi Gorantala
Posted on فبراير ١٨ • Updated on فبراير ٢١

Monotonic Array - Java Solution

#faang #leetcode #algorithms #programming

Introduction

A monotonic Array is an array if it is either increasing or monotonic decreasing.

Problem Statement

An array `A` is monotone increasing if for all `i <= j, A[i] <= A[j].`

An array `A` is monotone decreasing if for all `i <= j, A[i] >= A[j],` return `true` if and only if the given array `A` is monotonic.

Example 1:

```
Input: [1, 2, 2, 3]

Output: true
```

Example 2:

```
Input: [1, 2, 6, 2, 3]

Output: false
```

Example 3:

```
Input: [7, 2, 1]

Output: true
```

Thought Process

An `array` is called monotonic if the `index` of each element increases from the first to the last, or if its decreasing from the first to the last.

Algorithm

Algorithm

We need to run two for loops to check if either of the loops returns true.

For the monotonic increasing array, we need to check if the previous index is less than the current index. And for the monotonic decreasing array, we need to check if the previous index is greater than the current index.

Finally, we return `true` if either of the loops evaluates to `true`.

> Optimal way: we can do using one pass but let us take a look at both the algorithms.

Solution

Approach 01: Two-Pass Algorithm

```
public class MonotonicArray {
  public static void main(String[] args) {
    int[] input = {1, 2, 2, 3};
    System.out.println(isMonotonic(input)); // true
  }

  public static boolean isMonotonic(int[] array) {
    return isIncreasing(array) || isDecreasing(array);
  }

  public static boolean isIncreasing(int[] nums) {
    for (int i = 1; i < nums.length; i++)
      if (nums[i - 1] > nums[i]) {
        return false;
      }
    return true;
  }

  public static boolean isDecreasing(int[] nums) {
    for (int i = 1; i < nums.length; i++)
```

```
        if (nums[i - 1] < nums[i]) {
          return false;

        }
      return true;
    }
  }
```

Complexity Analysis

We are doing two for loops above and the overall time complexity is `O(n)`, and space complexity is `O(1)`.

Let us optimize the above code snippet with a single loop.

Approach 02: One-Pass Algorithm

```
public class MonotonicArray {
  public static void main(String[] args) {
    int[] input = {1, 2, 2, 3};
    System.out.println(isMonotonic(input)); // true
  }

  public static boolean isMonotonic(int[] array) {
    boolean isIncreasing = true;
    boolean isDecreasing = true;

    for (int i = 1; i < array.length; i++) {
      if (array[i] < array[i - 1]) {
        isDecreasing = false;
      }

      if (array[i] > array[i - 1]) {
        isIncreasing = false;
      }
    }
```

```
        }

        return isIncreasing || isDecreasing;
    }
}
```

Complexity Analysis

Overall complexity analysis won't change, but we are eliminating a loop here with this optimized approach.

Hence, the complexity analysis for both the approaches are time – `O(n)` and space – `O(1)`.

Extras

If you are interested in mastering bit tricks, I've got a course that are loved by more than 100k+ programmers.

In this course, you will learn how to solve problems using bit manipulation, a powerful technique that can be used to optimize your algorithmic and problem-solving skills. The course has simple explanation with sketches, detailed step-by-step drawings, and various ways to solve it using bitwise operators.

These bit-tricks could help in competitive programming and coding interviews in running algorithms mostly in `O(1)` time.

This is one of the most important/critical topics when someone starts preparing for coding interviews for FAANG(Facebook, Amazon, Apple, Netflix, and Google) companies.

To kick things off, you'll start by learning about the number system and how it's represented. Then you'll move on to learn about the six different bitwise operators: AND, OR, NOT, XOR, and bit shifting. Throughout, you will get tons of hands-on experience working through practice problems to help sharpen your understanding.

By the time you've completed this course, you will be able to solve problems faster with greater efficiency!! 😍

Link to my course: Master Bit Manipulation for Coding Interviews.

Discussion (0)

Code of Conduct • Report abuse

Gopi Gorantala

After working for European government projects in Belgium for half a decade, I decided to enter the life of a freelancer, I had roughly 10 years of experience under my belt --

LOCATION
Brussels, Belgium

EDUCATION
India

WORK
Checkout my Trending Course:- https://www.educative.io/courses/bit-manipulation

JOINED
١٨ فبراير ٢٠٢٢

More from Gopi Gorantala

What Are Java Method References And Kinds Of Method References Available?

#java #programming #beginners #tutorial

How Does Streams API Work? A Deep Dive on Stream Operation Flow

#java #programming #tutorial #beginners

Introduction To Java Streams API

#java #programming #tutorial #beginners