Problem 2016: Maximum Difference Between Increasing Elements

Problem Information

Difficulty: Easy
Acceptance Rate: 66.38%
Paid Only: No
Tags: Array

Problem Description

Given a **0-indexed** integer array `nums` of size `n`, find the **maximum difference** between `nums[i]` and `nums[j]` (i.e., `nums[j] - nums[i]`), such that `0 <= i < j < n` and `nums[i] < nums[j]`.

Return _the**maximum difference**. _If no such `i` and `j` exists, return `-1`.

Example 1:

Input: nums = [7,**_1_** ,**_5_** ,4] **Output:** 4 **Explanation:** The maximum difference occurs with i = 1 and j = 2, nums[j] - nums[i] = 5 - 1 = 4. Note that with i = 1 and j = 0, the difference nums[j] - nums[i] = 7 - 1 = 6, but i > j, so it is not valid.

Example 2:

Input: nums = [9,4,3,2] **Output:** -1 **Explanation:** There is no i and j such that i < j and nums[i] < nums[j].

Example 3:

Input: nums = [**_1_** ,5,2,**_10_**] **Output:** 9 **Explanation:** The maximum difference occurs with i = 0 and j = 3, nums[j] - nums[i] = 10 - 1 = 9.

Constraints:

* `n == nums.length` * `2 <= n <= 1000` * `1 <= nums[i] <= 109`

Code Snippets

C++:

```
class Solution {
public:
int maximumDifference(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int maximumDifference(int[] nums) {

}
}
```

Python3:

```
class Solution:
def maximumDifference(self, nums: List[int]) -> int:
```