1137 Height Checker (link)

Description

A school is trying to take an annual photo of all the students. The students are asked to stand in a single file line in **non-decreasing order** by height. Let this ordering be represented by the integer array `expected` where `expected[i]` is the expected height of the `i`$^{\text{th}}$ student in line.

You are given an integer array `heights` representing the **current order** that the students are standing in. Each `heights[i]` is the height of the `i`$^{\text{th}}$ student in line (**0-indexed**).

Return *the* ***number of indices*** *where* `heights[i] != expected[i]`.

Example 1:

```
Input: heights = [1,1,4,2,1,3]
Output: 3
Explanation:
heights:  [1,1,4,2,1,3]
expected: [1,1,1,2,3,4]
Indices 2, 4, and 5 do not match.
```

Example 2:

```
Input: heights = [5,1,2,3,4]
Output: 5
Explanation:
heights:  [5,1,2,3,4]
expected: [1,2,3,4,5]
All indices do not match.
```

Example 3:

```
Input: heights = [1,2,3,4,5]
Output: 0
Explanation:
heights:  [1,2,3,4,5]
expected: [1,2,3,4,5]
All indices match.
```

Constraints:

- `1 <= heights.length <= 100`
- `1 <= heights[i] <= 100`

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```cpp
#include <vector>
#include <algorithm>

using namespace std;

class Solution {
public:
    int heightChecker(vector<int>& heights) {
        vector<int> expected = heights;
        sort(expected.begin(), expected.end()); // Ожидаемые значения - отсортированны

        int mismatch_count = 0;
        for (int i = 0; i < heights.size(); ++i) {
            if (heights[i] != expected[i]) {
                mismatch_count++;
            }
        }

        return mismatch_count;
    }
};
```