Height Checker

# Question

Students are asked to stand in non-decreasing order of heights for an annual photo.

Return the minimum number of students that must move in order for all students to be standing in non-decreasing order of height.

Notice that when a group of students is selected they can reorder in any possible way between themselves and the non selected students remain on their seats.

**Example 1:**

Input: heights = [1,1,4,2,1,3]

Output: 3

Explanation:

Current array : [1,1,4,2,1,3]

Target array : [1,1,1,2,3,4]

On index 2 (0-based) we have 4 vs 1 so we have to move this student.

On index 4 (0-based) we have 1 vs 3 so we have to move this student.

On index 5 (0-based) we have 3 vs 4 so we have to move this student.

**Example 2:**

Input: heights = [5,1,2,3,4]

Output: 5

**Example 3:**

Input: heights = [1,2,3,4,5]

Output: 0

# Pseudo Code

Create a new array and copy the contents from the Original Array

Sort the New Array

Declare minimumStudents and Initialize it to 0

Run the For Loop

If Elements are not equal

Increase the minimumStudents by one

Return minimumStudents

# Source Code

## v1.0 (Language C)

1. int heightChecker(int\* heights, int heightsSize){
3. int sortedHeights[heightsSize], tempVariable = 0, minimumStudents = 0;
5. if(heightsSize < 2)
6. return 0;
8. for(int i=0 ; i<heightsSize ; i++) {
9. sortedHeights[i] = heights[i];
10. }
12. for(int i=0 ; i<(heightsSize - 1) ; i++) {
14. for(int j=0 ; j<(heightsSize - i - 1) ; j++) {
15. if(sortedHeights[j + 1] < sortedHeights[j]) {
16. tempVariable = sortedHeights[j];
17. sortedHeights[j] = sortedHeights[j + 1];
18. sortedHeights[j + 1] = tempVariable;
19. }
20. }
21. }
23. for(int i=0 ; i<heightsSize ; i++) {
24. if(heights[i] != sortedHeights[i]) {
25. minimumStudents++;
26. }
27. }
28. return minimumStudents;
29. }