Problem 853: Car Fleet

Problem Information

Difficulty: Medium
Acceptance Rate: 54.27%
Paid Only: No
Tags: Array, Stack, Sorting, Monotonic Stack

Problem Description

There are `n` cars at given miles away from the starting mile 0, traveling to reach the mile `target`.

You are given two integer arrays `position` and `speed`, both of length `n`, where `position[i]` is the starting mile of the `ith` car and `speed[i]` is the speed of the `ith` car in miles per hour.

A car cannot pass another car, but it can catch up and then travel next to it at the speed of the slower car.

A **car fleet** is a single car or a group of cars driving next to each other. The speed of the car fleet is the **minimum** speed of any car in the fleet.

If a car catches up to a car fleet at the mile `target`, it will still be considered as part of the car fleet.

Return the number of car fleets that will arrive at the destination.

Example 1:

Input: target = 12, position = [10,8,0,5,3], speed = [2,4,1,1,3]

Output: 3

Explanation:

* The cars starting at 10 (speed 2) and 8 (speed 4) become a fleet, meeting each other at 12. The fleet forms at `target`. * The car starting at 0 (speed 1) does not catch up to any other car, so it is a fleet by itself. * The cars starting at 5 (speed 1) and 3 (speed 3) become a fleet, meeting each other at 6. The fleet moves at speed 1 until it reaches `target`.

Example 2:

Input: target = 10, position = [3], speed = [3]

Output: 1

Explanation:

There is only one car, hence there is only one fleet.

Example 3:

Input: target = 100, position = [0,2,4], speed = [4,2,1]

Output: 1

Explanation:

* The cars starting at 0 (speed 4) and 2 (speed 2) become a fleet, meeting each other at 4. The car starting at 4 (speed 1) travels to 5. * Then, the fleet at 4 (speed 2) and the car at position 5 (speed 1) become one fleet, meeting each other at 6. The fleet moves at speed 1 until it reaches `target`.

Constraints:

* `n == position.length == speed.length` * `1 <= n <= 105` * `0 < target <= 106` * `0 <= position[i] < target` * All the values of `position` are **unique**. * `0 < speed[i] <= 106`

Code Snippets

C++:

```
class Solution {
public:
```

```
int carFleet(int target, vector<int>& position, vector<int>& speed) {

}
};
```

Java:

```
class Solution {
public int carFleet(int target, int[] position, int[] speed) {

}
}
```

Python3:

```
class Solution:
def carFleet(self, target: int, position: List[int], speed: List[int]) ->
int:
```