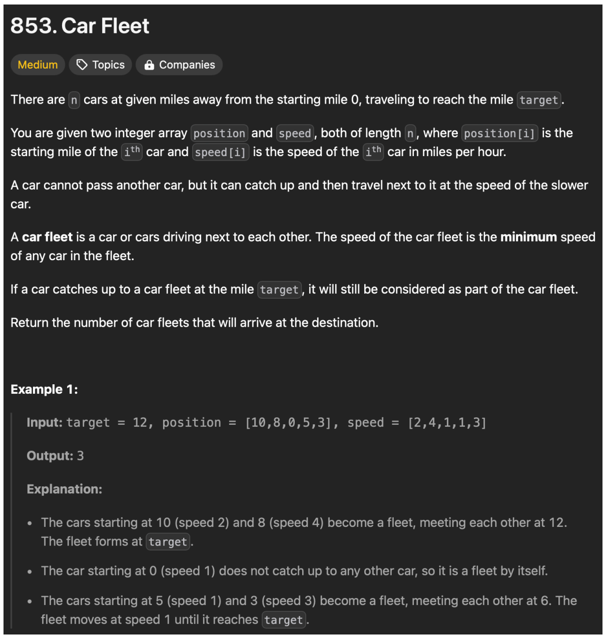


853. Car Fleet

Medium · Topics · Companies

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

You are given two integer array `position` and `speed`, both of length `n`, where `position[i]` is the starting mile of the i^{th} car and `speed[i]` is the speed of the i^{th} 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.

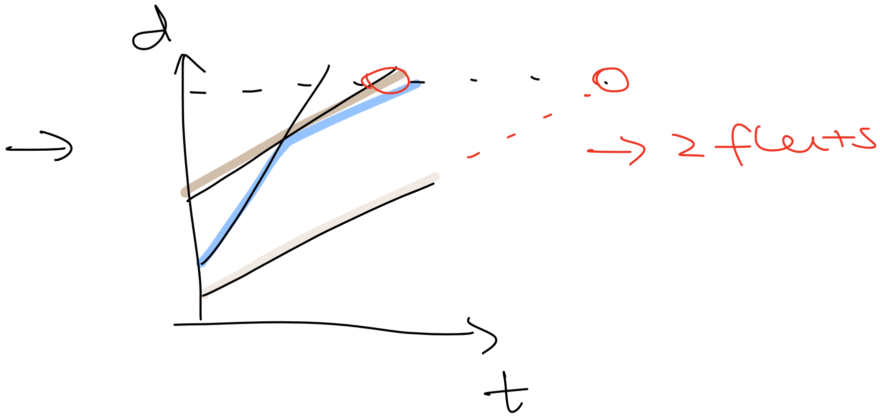
A **car fleet** is a car or cars driving next to each other. The speed of the car fleet is the **minimum** speed of any car in the fleet.

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`.



$$\frac{12-10}{2} = 1$$

$$\frac{12-8}{4} = 1$$

$$\frac{12-5}{1} = 7$$

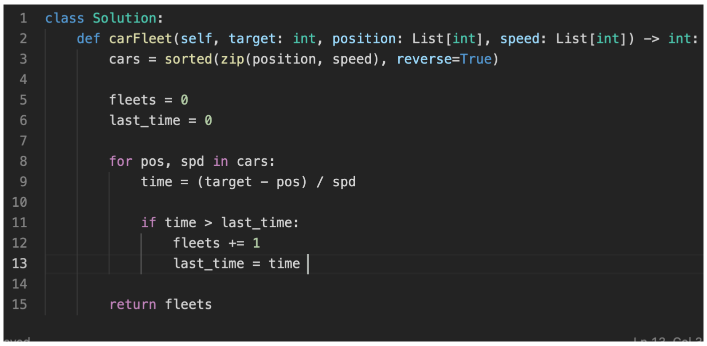
$$\frac{12-3}{3} = 3$$

$$\frac{12-0}{1} = 12$$

sort by distance

if $time_i > time_{i-1}$:

new fleet



```
class Solution:
    def carFleet(self, target: int, position: List[int], speed: List[int]) -> int:
        cars = sorted(zip(position, speed), reverse=True)

        fleets = 0
        last_time = 0

        for pos, spd in cars:
            time = (target - pos) / spd

            if time > last_time:
                fleets += 1
                last_time = time

        return fleets
```