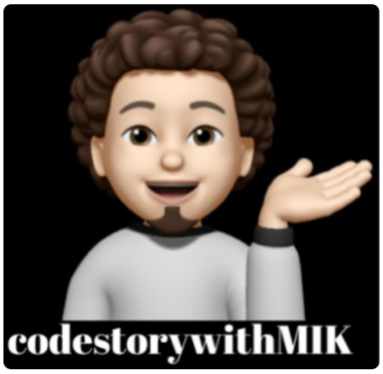
GREEDY



VIDEO - 21

LEETCODE - 1921

Medium

Easy बना देंगे

Eliminate Maximum Number of Monsters

Facebook
Instagram → codestorywithMIK

Twith → cswithMIK

→ codestorywithMIK

Company Tags :- amazon Meta Google

1921. Eliminate Maximum Number of Monsters

Hint

Medium 624 99

Companies

You are playing a video game where you are defending your city from a group of `n` monsters. You are given a **0-indexed** integer array `dist` of size `n`, where `dist[i]` is the **initial distance** in kilometers of the `i^th` monster from the city.

The monsters walk toward the city at a **constant** speed. The speed of each monster is given to you in an integer array `speed` of size `n`, where `speed[i]` is the speed of the `i^th` monster in kilometers per minute.

You have a weapon that, once fully charged, can eliminate a **single** monster. However, the weapon takes **one minute** to charge. The weapon is fully charged at the very start.

You lose when any monster reaches your city. If a monster reaches the city at the exact moment the weapon is fully charged, it counts as a **loss**, and the game ends before you can use your weapon.

Return *the **maximum** number of monsters that you can eliminate before you lose, or* `n` *if you can eliminate all the monsters before they reach the city.*

Example :-

m1 m2 m3

dist = {1, 3, 4}

speed = {1, 1, 1}

charged = 1 1 1

3

dist = {1, 2, 3}
speed = {1, 1, 1}

charged = 1

1

dist = {3, 4}
speed = {5, 3, 2}

charged = 1

1

1

Intuition :-

time.

" मैं मजबूर हूँ 1 minute

के time से — Gun reload "

For me, most important thing

is time.

"I will kill the monster which is the FASTEST → one who will reach city in least time."

M1 M2

{100, 5}

{100, 1}

$time_{(M1)} = \frac{100}{100} = 1$

$time_{(M2)} = \frac{5}{1} = 5$

(•) time → monsters.

(•) Sort the time according to ascending order.

dist = {1, 3, 4}

speed = {1, 1, 1}

time = {~~X~~, 3, 4}

time = 1+1

~~2~~ ~~2~~

Count = 1+1+1

dist = {3, 2, 4}

speed = {5, 3, 2}

time = {1, 1, 2}

time_pass = 0+1

Count = 1

- time → sort
- Count = 1
- time_passed = 1

```
for ( i = 1 ; i<n ; i++) {
    if(time[i] - time_passed <=0){
        return count ;
    }
    Count += 1 ;
    time_passed += 1 ;
}
return Count ;
```