

Problem 3207: Maximum Points After Enemy Battles

Problem Information

Difficulty: Medium

Acceptance Rate: 32.93%

Paid Only: No

Tags: Array, Greedy

Problem Description

You are given an integer array `enemyEnergies` denoting the energy values of various enemies.

You are also given an integer `currentEnergy` denoting the amount of energy you have initially.

You start with 0 points, and all the enemies are unmarked initially.

You can perform **either** of the following operations **zero** or multiple times to gain points:

* Choose an **unmarked** enemy, `i`, such that `currentEnergy >= enemyEnergies[i]`. By choosing this option: * You gain 1 point. * Your energy is reduced by the enemy's energy, i.e. `currentEnergy = currentEnergy - enemyEnergies[i]`. * If you have **at least** 1 point, you can choose an **unmarked** enemy, `i`. By choosing this option: * Your energy increases by the enemy's energy, i.e. `currentEnergy = currentEnergy + enemyEnergies[i]`. * The enemy `i` is **marked**.

Return an integer denoting the **maximum** points you can get in the end by optimally performing operations.

Example 1.

Input: `enemyEnergies = [3,2,2], currentEnergy = 2`

Output: 3

****Explanation:****

The following operations can be performed to get 3 points, which is the maximum:

* First operation on enemy 1: `points` increases by 1, and `currentEnergy` decreases by 2. So, `points = 1`, and `currentEnergy = 0`. * Second operation on enemy 0: `currentEnergy` increases by 3, and enemy 0 is marked. So, `points = 1`, `currentEnergy = 3`, and marked enemies = `[0]`. * First operation on enemy 2: `points` increases by 1, and `currentEnergy` decreases by 2. So, `points = 2`, `currentEnergy = 1`, and marked enemies = `[0]`. * Second operation on enemy 2: `currentEnergy` increases by 2, and enemy 2 is marked. So, `points = 2`, `currentEnergy = 3`, and marked enemies = `[0, 2]`. * First operation on enemy 1: `points` increases by 1, and `currentEnergy` decreases by 2. So, `points = 3`, `currentEnergy = 1`, and marked enemies = `[0, 2]`.

****Example 2:****

****Input:**** enemyEnergies = [2], currentEnergy = 10

****Output:**** 5

****Explanation:****

Performing the first operation 5 times on enemy 0 results in the maximum number of points.

****Constraints:****

* `1 <= enemyEnergies.length <= 105` * `1 <= enemyEnergies[i] <= 109` * `0 <= currentEnergy <= 109`

Code Snippets

C++:

```
class Solution {
public:
    long long maximumPoints(vector<int>& enemyEnergies, int currentEnergy) {

    }
};
```

Java:

```
class Solution {  
    public long maximumPoints(int[] enemyEnergies, int currentEnergy) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def maximumPoints(self, enemyEnergies: List[int], currentEnergy: int) -> int:
```