Problem 3273: Minimum Amount of Damage Dealt to Bob

Problem Information

Difficulty: Hard
Acceptance Rate: 38.99%
Paid Only: No
Tags: Array, Greedy, Sorting

Problem Description

You are given an integer `power` and two integer arrays `damage` and `health`, both having length `n`.

Bob has `n` enemies, where enemy `i` will deal Bob `damage[i]` **points** of damage per second while they are _alive_ (i.e. `health[i] > 0`).

Every second, **after** the enemies deal damage to Bob, he chooses **one** of the enemies that is still _alive_ and deals `power` points of damage to them.

Determine the **minimum** total amount of damage points that will be dealt to Bob before **all** `n` enemies are _dead_.

Example 1:

Input: power = 4, damage = [1,2,3,4], health = [4,5,6,8]

Output: 39

Explanation:

* Attack enemy 3 in the first two seconds, after which enemy 3 will go down, the number of damage points dealt to Bob is `10 + 10 = 20` points. * Attack enemy 2 in the next two seconds, after which enemy 2 will go down, the number of damage points dealt to Bob is `6 + 6 = 12` points. * Attack enemy 0 in the next second, after which enemy 0 will go down, the number of damage points dealt to Bob is `3` points. * Attack enemy 1 in the next two seconds,

after which enemy 1 will go down, the number of damage points dealt to Bob is `2 + 2 = 4` points.

Example 2:

Input: power = 1, damage = [1,1,1,1], health = [1,2,3,4]

Output: 20

Explanation:

* Attack enemy 0 in the first second, after which enemy 0 will go down, the number of damage points dealt to Bob is `4` points. * Attack enemy 1 in the next two seconds, after which enemy 1 will go down, the number of damage points dealt to Bob is `3 + 3 = 6` points. * Attack enemy 2 in the next three seconds, after which enemy 2 will go down, the number of damage points dealt to Bob is `2 + 2 + 2 = 6` points. * Attack enemy 3 in the next four seconds, after which enemy 3 will go down, the number of damage points dealt to Bob is `1 + 1 + 1 + 1 = 4` points.

Example 3:

Input: power = 8, damage = [40], health = [59]

Output: 320

Constraints:

* `1 <= power <= 104` * `1 <= n == damage.length == health.length <= 105` * `1 <= damage[i], health[i] <= 104`

Code Snippets

C++:

```
class Solution {
public:
long long minDamage(int power, vector<int>& damage, vector<int>& health) {

}
};
```

Java:

```
class Solution {
public long minDamage(int power, int[] damage, int[] health) {

}
}
```

Python3:

```
class Solution:
def minDamage(self, power: int, damage: List[int], health: List[int]) -> int:
```