6075. Maximum Bags With Full Capacity of Rocks

My Submissions (/contest/weekly-contest-294/problems/maximum-bags-with-full-capacity-of-rocks/submissions/)

Back to Contest (/contest/weekly-contest-294/)

You have `n` bags numbered from `0` to `n - 1`. You are given two **0-indexed** integer arrays `capacity` and `rocks`. The `i`th bag can hold a maximum of `capacity[i]` rocks and currently contains `rocks[i]` rocks. You are also given an integer `additionalRocks`, the number of additional rocks you can place in **any** of the bags.

Return *the **maximum** number of bags that could have full capacity after placing the additional rocks in some bags.*

User Accepted:	2282
User Tried:	2494
Total Accepted:	2286
Total Submissions:	2682
Difficulty:	Medium

Example 1:

```
Input: capacity = [2,3,4,5], rocks = [1,2,4,4], additionalRocks = 2
Output: 3
Explanation:
Place 1 rock in bag 0 and 1 rock in bag 1.
The number of rocks in each bag are now [2,3,4,4].
Bags 0, 1, and 2 have full capacity.
There are 3 bags at full capacity, so we return 3.
It can be shown that it is not possible to have more than 3 bags at full capacity.
Note that there may be other ways of placing the rocks that result in an answer of 3.
```

Example 2:

```
Input: capacity = [10,2,2], rocks = [2,2,0], additionalRocks = 100
Output: 3
Explanation:
Place 8 rocks in bag 0 and 2 rocks in bag 2.
The number of rocks in each bag are now [10,2,2].
Bags 0, 1, and 2 have full capacity.
There are 3 bags at full capacity, so we return 3.
It can be shown that it is not possible to have more than 3 bags at full capacity.
Note that we did not use all of the additional rocks.
```

Constraints:

- `n == capacity.length == rocks.length`
- `1 <= n <= 5 * 10^4`
- `1 <= capacity[i] <= 10^9`
- `0 <= rocks[i] <= capacity[i]`
- `1 <= additionalRocks <= 10^9`

JavaScript

```javascript
const maximumBags = (capacity, rocks, k) => {
    let n = rocks.length, d = [], res = 0;
    for (let i = 0; i < n; i++) d.push([rocks[i], capacity[i], capacity[i] - rocks[i]]);
    d.sort((x, y) => x[2] - y[2]);
    for (const [x, y, need] of d) {
        if (need == 0) {
            res++;
        } else {
            if (k >= need) {
                k -= need;
                res++;
            } else {
                break;
            }
        }
    }
    return res;
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/704457382/) More Details (/submissions/detail/704457382/)

Share your acceptance!

2