Problem 2558: Take Gifts From the Richest Pile

Problem Information

Difficulty: Easy
Acceptance Rate: 75.54%
Paid Only: No
Tags: Array, Heap (Priority Queue), Simulation

Problem Description

You are given an integer array `gifts` denoting the number of gifts in various piles. Every second, you do the following:

* Choose the pile with the maximum number of gifts. * If there is more than one pile with the maximum number of gifts, choose any. * Reduce the number of gifts in the pile to the floor of the square root of the original number of gifts in the pile.

Return _the number of gifts remaining after_`k` _seconds._

Example 1:

Input: gifts = [25,64,9,4,100], k = 4 **Output:** 29 **Explanation:** The gifts are taken in the following way: - In the first second, the last pile is chosen and 10 gifts are left behind. - Then the second pile is chosen and 8 gifts are left behind. - After that the first pile is chosen and 5 gifts are left behind. - Finally, the last pile is chosen again and 3 gifts are left behind. The final remaining gifts are [5,8,9,4,3], so the total number of gifts remaining is 29.

Example 2:

Input: gifts = [1,1,1,1], k = 4 **Output:** 4 **Explanation:** In this case, regardless which pile you choose, you have to leave behind 1 gift in each pile. That is, you can't take any pile with you. So, the total gifts remaining are 4.

Constraints:

* `1 <= gifts.length <= 103` * `1 <= gifts[i] <= 109` * `1 <= k <= 103`

Code Snippets

C++:

```
class Solution {
public:
long long pickGifts(vector<int>& gifts, int k) {

}
};
```

Java:

```
class Solution {
public long pickGifts(int[] gifts, int k) {

}
}
```

Python3:

```
class Solution:
def pickGifts(self, gifts: List[int], k: int) -> int:
```