

# Problem 2137: Pour Water Between Buckets to Make Water Levels Equal

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 67.52%

**Paid Only:** Yes

**Tags:** Array, Binary Search

## Problem Description

You have `n` buckets each containing some gallons of water in it, represented by a \*\*0-indexed\*\* integer array `buckets`, where the `ith` bucket contains `buckets[i]` gallons of water. You are also given an integer `loss`.

You want to make the amount of water in each bucket equal. You can pour any amount of water from one bucket to another bucket (not necessarily an integer). However, every time you pour `k` gallons of water, you spill `loss` \*\*percent\*\* of `k`.

Return \_the\*\*maximum\*\* amount of water in each bucket after making the amount of water equal. \_Answers within `10-5` of the actual answer will be accepted.

\*\*Example 1:\*\*

\*\*Input:\*\* buckets = [1,2,7], loss = 80 \*\*Output:\*\* 2.00000 \*\*Explanation:\*\* Pour 5 gallons of water from buckets[2] to buckets[0].  $5 * 80\% = 4$  gallons are spilled and buckets[0] only receives  $5 - 4 = 1$  gallon of water. All buckets have 2 gallons of water in them so return 2.

\*\*Example 2:\*\*

\*\*Input:\*\* buckets = [2,4,6], loss = 50 \*\*Output:\*\* 3.50000 \*\*Explanation:\*\* Pour 0.5 gallons of water from buckets[1] to buckets[0].  $0.5 * 50\% = 0.25$  gallons are spilled and buckets[0] only receives  $0.5 - 0.25 = 0.25$  gallons of water. Now, buckets = [2.25, 3.5, 6]. Pour 2.5 gallons of water from buckets[2] to buckets[0].  $2.5 * 50\% = 1.25$  gallons are spilled and buckets[0] only receives  $2.5 - 1.25 = 1.25$  gallons of water. All buckets have 3.5 gallons of water in them so return 3.5.

**\*\*Example 3:\*\***

**\*\*Input:\*\*** buckets = [3,3,3,3], loss = 40   **\*\*Output:\*\*** 3.00000   **\*\*Explanation:\*\*** All buckets already have the same amount of water in them.

**\*\*Constraints:\*\***

\* `1 <= buckets.length <= 105` \* `0 <= buckets[i] <= 105` \* `0 <= loss <= 99`

## Code Snippets

**C++:**

```
class Solution {  
public:  
    double equalizeWater(vector<int>& buckets, int loss) {  
  
    }  
};
```

**Java:**

```
class Solution {  
public double equalizeWater(int[] buckets, int loss) {  
  
}  
}
```

**Python3:**

```
class Solution:  
    def equalizeWater(self, buckets: List[int], loss: int) -> float:
```