

756. Pour Water

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Solution](#)

- Difficulty:Medium
- Total Accepted:1.1K
- Total Submissions:3.2K
- Contributor:[1337c0d3r](#)
-
- [Subscribe](#) to see which companies asked this question.

Related Topics [Array](#)

We are given an elevation map, `heights[i]` representing the height of the terrain at that index. The width at each index is 1. After V units of water fall at index K , how much water is at each index?

Water first drops at index K and rests on top of the highest terrain or water at that index. Then, it flows according to the following rules:

- If the droplet would eventually fall by moving left, then move left.
- Otherwise, if the droplet would eventually fall by moving right, then move right.
- Otherwise, rise at it's current position. Here, "eventually fall" means that the droplet will eventually be at a lower level if it moves in that direction. Also, "level" means the height of the terrain plus any water in that column.

We can assume there's infinitely high terrain on the two sides out of bounds of the array. Also, there could not be partial water being spread out evenly on more than 1 grid block - each unit of water has to be in exactly one block.

Example 1:

Input: `heights = [2,1,1,2,1,2,2]`, $V = 4$, $K = 3$

Output: `[2,2,2,3,2,2,2]`

Explanation:

```
#          #
#          #
##  #  ###
#####
0123456    <- index
```

The first drop of water lands at index $K = 3$:

```
#          #
#   w   #
##  #  ###
#####
0123456
```

When moving left or right, the water can only move to the same level or a lower level.

(By level, we mean the total height of the terrain plus any water in that column.)
 Since moving left will eventually make it fall, it moves left.
 (A droplet "made to fall" means go to a lower height than it was at previously.)

```
#      #
#      #
## w# ###
#####
0123456
```

Since moving left will not make it fall, it stays in place. The next droplet falls:

```
#      #
#  w   #
## w# ###
#####
0123456
```

Since the new droplet moving left will eventually make it fall, it moves left.
 Notice that the droplet still preferred to move left,
 even though it could move right (and moving right makes it fall quicker.)

```
#      #
#  w   #
## w# ###
#####
0123456
```

```
#      #
#      #
##ww# ###
#####
0123456
```

After those steps, the third droplet falls.
 Since moving left would not eventually make it fall, it tries to move right.
 Since moving right would eventually make it fall, it moves right.

```
#      #
#  w   #
##ww# ###
#####
0123456
```

```
#      #
#      #
##ww#w###
#####
0123456
```

Finally, the fourth droplet falls.
 Since moving left would not eventually make it fall, it tries to move right.
 Since moving right would not eventually make it fall, it stays in place:

```
#      #
#  w   #
##ww#w###
#####
0123456
```

The final answer is [2,2,2,3,2,2,2]:

```
#
#####
#####
0123456
```

Example 2:

Input: heights = [1,2,3,4], V = 2, K = 2

Output: [2,3,3,4]

Explanation:

The last droplet settles at index 1, since moving further left would not cause it to eventually fall to a lower height.

Example 3:

Input: heights = [3,1,3], V = 5, K = 1

Output: [4,4,4]

Note:

1. heights will have length in [1, 100] and contain integers in [0, 99].
2. V will be in range [0, 2000].
3. K will be in range [0, heights.length - 1].

Simple C++ Solution

```
vector<int> pourWater(vector<int>& heights, int V, int K) {
    int index;
    while(V)
    {
        index = K;
        for(int i=K-1;i>=0;i--)
        {
            if(heights[i]>heights[index])
                break;
            else if (heights[i]<heights[index]){
                index=i;
            }
        }
        if(index!=K)
        {
            heights[index]++;
            V--;
            continue;
        }
        for(int i=K+1;i<(int)heights.size();i++)
        {
            if(heights[i]>heights[index])
                break;
            else if (heights[i]<heights[index]){
                index=i;
            }
        }
        heights[index]++;
        V--;
    }
}
```

```
    return heights;  
}
```