題目 577

✔577. 淹水啦！！！

Submit | Status | Discussion / Solution | Ranklist | Back to Problems List

TopCoder

abcabcabc

有人要寫 p6 嗎 > <

User's AC Ratio: 57.1% (4/7)

Submission's AC Ratio: 21.7% (5/23)

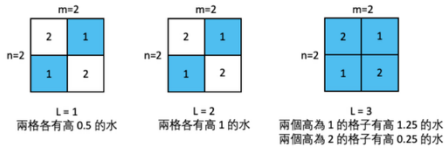
Tags: Show solution-related tags — apcsc | apcsc-2darray

Description

學完了二維陣列後，小明興高采烈地回家，書包往沙發上一丟就馬上打開他的電腦。點了兩下螢幕上的草地方塊，打開了某款小明最喜歡的沙盒遊戲。這款沙盒遊戲最大的特點就是整張地圖皆是由一個個方塊所構成！

但那天小明發現他在遊戲中的家被調皮的朋友倒了水進去，整個家充滿了一坑坑的水窪。生氣的小明決定向那位調皮的朋友索賠，依被水蓋過的土地面積計價！

已知小明的家為一個 $n \times m$ 的矩形，給定家中每格對應的高度，第 i 列第 j 行的那格所對應的高度為 $h_{i,j}(0 \le i < n, 0 \le j < m)$。調皮的朋友倒了 L 單位的水進去，假設水會先平均分佈於最低的那層，如果最低的那層被水淹滿後還有多餘的水，就會再平均覆蓋在次低的那層。如圖所示，方格內顯示的數字是該格的高度：



請你幫小明寫個程式計算淹水的區域。

Input Format

輸入總共有 $1+n$ 行。第一行包含三個以單一空格隔開的正整數 n, m, L，代表小明家的長寬，以及朋友倒入 L 單位的水。接下來有 n 行，每行有 m 個整數 $h_{i,j}$ 代表該格的高度。

- $1 \le n, m \le 100$
- $0 \le L \le 2^{31} - 1$
- $0 \le h_{i,j} \le 100$

Output Format

請輸出整張 $n \times m$ 的地圖。輸出共有 n 行，每行有 m 個整數 `0` 或 `1` 代表該格淹水的情況。若 $h_{i,j}$ 被水淹過請輸出 `1`，否則請輸出 `0`。

Sample Input 1

```
4 4 0
2 1 2 0
1 0 1 1
0 1 4 7
2 3 0 1
```

Sample Output 1

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

Sample Input 2

```
2 2 10
0 1
1 0
```

Sample Output 2

```
1 1
1 1
```

這一題我原本的想法是將所有高度加總再加上倒入水量後取平均，如果該地點的高度低於平均就會淹水

Problem	Total Time (ms)	Max Memory (RSS, KiB)	Verdict	Score
577. 淹水啦！！！	116	3456	Wrong Answer	0

Subtask Results

Subtask no.	Testdata Range	Constraints	Score
1	0–1	範例測資	0 / 0
2	0–16	無額外限制	0 / 100

Testdata Results

Testdata no.	Subtasks	Time (ms)	Memory (VSS, KiB)	Memory (RSS, KiB)	Verdict	Score
0	1 2	3.3	6356	3456	Accepted	100
1	1 2	4.1	6356	3456	Accepted	100
2	2	4.1	6356	3456	Wrong Answer	0
3	2	4.1	6356	3456	Wrong Answer	0
4	2	4.2	6356	3456	Wrong Answer	0
5	2	4.2	6356	3456	Wrong Answer	0
6	2	4.3	6356	3456	Wrong Answer	0
7	2	4.2	6356	3456	Wrong Answer	0
8	2	4.3	6356	3456	Wrong Answer	0
9	2	5.6	6356	3456	Wrong Answer	0
10	2	6.9	6356	3456	Wrong Answer	0
11	2	11.1	6356	3456	Wrong Answer	0
12	2	11.4	6356	3456	Wrong Answer	0
13	2	11.0	6356	3456	Wrong Answer	0
14	2	11.1	6356	3456	Wrong Answer	0
15	2	11.1	6356	3456	Wrong Answer	0
16	2	11.1	6356	3456	Wrong Answer	0

```cpp
#include <iostream>
using namespace std;
int main(){
    int n, m;
    int L;
    cin >> n >> m >> L;
    int **arr = new int *[n];
    for (int i = 0; i < n; i++)
    {
        arr[i] = new int [m];
    }
    int sum = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cin >> arr[i][j];
            sum += arr[i][j];
        }

    }
    if (L == 0){
        for (int i = 0; i < n; i++){
            for (int j = 0; j < m; j++){
                cout << 0 << " ";
            }
            cout << endl;
        }
        return 0;
    }else{
        L /= (n * m);
        sum += L;
    }
    int avg = sum / (m * n);
    for (int i = 0; i < n; i++){
            for (int j = 0; j < m; j++)
            {
                cout << avg << " ";
```

```
            }
            cout << endl;
    }
}
```

但這樣不是正確的解題思路， 一樣沒有考慮到極端值，且水應該是從低處開始填滿

Problem	Total Time (ms)	Max Memory (RSS, KiB)	Verdict	Score
577. 淹水啦！！！	2482	3456	Accepted	100

Subtask Results

Subtask no.	Testdata Range	Constraints	Score
1	0–1	範例測資	0 / 0
2	0–16	無額外限制	100 / 100

Testdata Results

Testdata no.	Subtasks	Time (ms)	Memory (VSS, KiB)	Memory (RSS, KiB)	Verdict	Score
0	1 2	3.0	6356	3456	Accepted	100
1	1 2	4.0	6356	3456	Accepted	100
2	2	4.1	6356	3456	Accepted	100
3	2	4.1	6356	3456	Accepted	100
4	2	4.1	6356	3456	Accepted	100
5	2	4.3	6356	3456	Accepted	100
6	2	4.7	6356	3456	Accepted	100
7	2	5.3	6356	3456	Accepted	100
8	2	6.7	6356	3456	Accepted	100
9	2	40.4	6356	3456	Accepted	100
10	2	72.9	6356	3456	Accepted	100
11	2	382.4	6512	3456	Accepted	100
12	2	387.2	6512	3456	Accepted	100
13	2	392.4	6512	3456	Accepted	100
14	2	391.8	6512	3456	Accepted	100
15	2	386.9	6512	3456	Accepted	100
16	2	388.1	6512	3456	Accepted	100

Code

```cpp

#include <iostream>
using namespace std;
int main()
{
    int N, M;
    cin >> N >> M;
    long long L;
    cin >> L;
    int **arr = new int*[N];
    for(int i = 0;i < N;i++){
        arr[i] = new int [M];
    }
    for(int i = 0;i < N;i++){
        for(int j = 0;j < M;j++){
            cin >> arr[i][j];
        }
    }
    int *sub = new int [(N * M)];
    int idx = 0;
    for(int i = 0;i < N;i++){
        for(int j = 0;j < M;j++){
            sub[idx] = arr[i][j];
            idx++;
        }
    }
    for(int i = 0;i < idx;i++){
        for(int j = 0;j < idx - 1;j++){
            if(sub[j] > sub[j + 1]){
                int a = sub[j];
                sub[j] = sub[j+1];
                sub[j + 1] = a;
            }
        }
    }
    double level = sub[0];
    for(int i = 0;i < idx - 1;i++){
        long long diff = sub[i + 1] - sub[i];
        long long need = diff * (i + 1);
        if(L > need){
            L -= need;
            level = sub[i + 1];
        }else{
            level = sub[i] + (double)L / (i + 1);
            L = 0;
            break;
        }
    }
    if (L > 0)
    {
        level += (double)L / (idx);
    }
    for(int i = 0;i < N;i++){
        for(int j = 0;j < M;j++){
            if(arr[i][j] < level){
                cout << 1 << " ";
            }else{
                cout << 0 << " ";
            }
        }
        cout << endl;
    }
    return 0;
}
```

```
int *sub = new int [(N * M)];
int idx = 0;
for(int i = 0;i < N;i++){
        for(int j = 0;j < M;j++){
        sub[idx] = arr[i][j];
        idx++;
        }
}
for(int i = 0;i < idx;i++){
        for(int j = 0;j < idx - 1;j++){
                if(sub[j] > sub[j + 1]){
                    int a = sub[j];
                    sub[j] = sub[j+1];
                    sub[j + 1] = a;
                }
        }
}
```

我改善的做法是先將二維陣列轉成一維後由小排到大

```
double level = sub[0];            //設置水面為浮點數，從最小的開始
for(int i = 0;i < idx - 1;i++){
        long long diff = sub[i + 1] - sub[i];
        long long need = diff * (i + 1);//消耗的水量為前後台階的高度差乘目前已經淹過的台階數量
        if(L > need){
                L -= need;
                level = sub[i + 1];
        }else{
                level = sub[i] + (double)L / (i + 1);
```

```
                L = 0;
                break;
        }
}
if (L > 0){
        level += (double)L / (idx);
}
for(int i = 0;i < N;i++){
        for(int j = 0;j < M;j++){
                if(arr[i][j] < level){
                        cout << 1 << " ";
                }else{
                        cout << 0 << " ";
                }
        }
        cout << endl;
}
```