407. Trapping Rain Water II

Given an m x n matrix of positive integers representing the height of each unit cell in a 2D elevation map, compute the volume of water it is able to trap after raining.

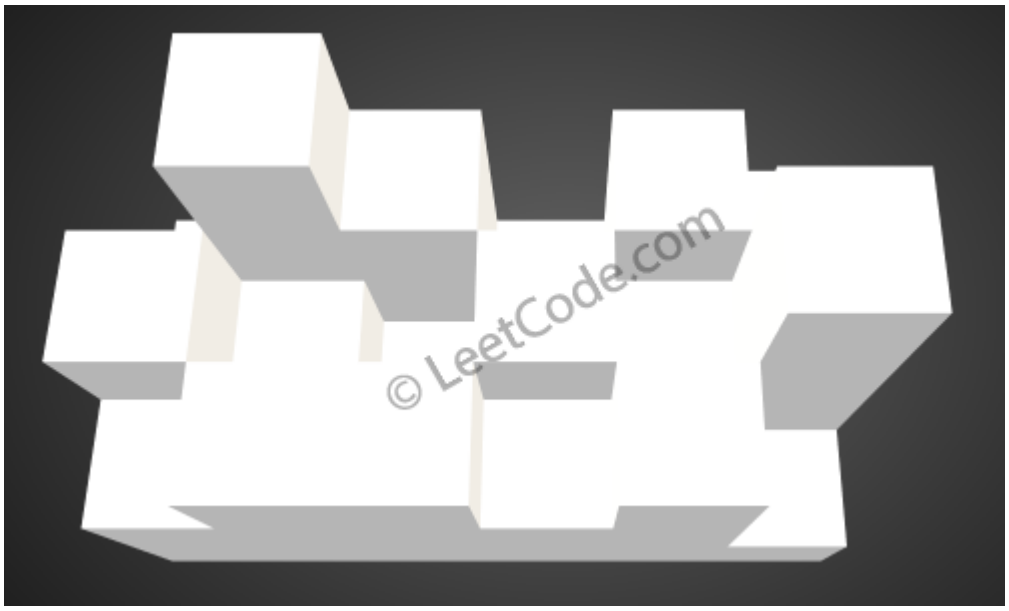
Note:

Both *m* and *n* are less than 110. The height of each unit cell is greater than 0 and is less than 20,000.

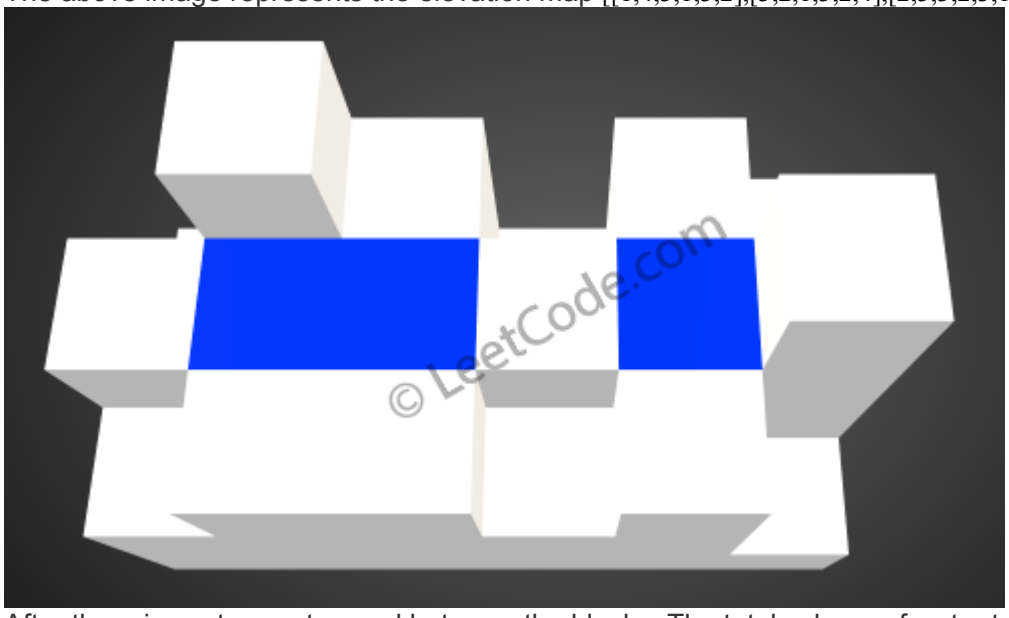
Given the following 3x6 height map:

```
[
  [1,4,3,1,3,2],
  [3,2,1,3,2,4],
  [2,3,3,2,3,1]
]
```

Return 4.



The above image represents the elevation map [[1,4,3,1,3,2],[3,2,1,3,2,4],[2,3,3,2,3,1]] before the rain.



After the rain, water are trapped between the blocks. The total volume of water trapped is 4.

```
#include<sstream>
#include<iostream>
#include<algorithm>
#include<string>
#include<vector>
#include<map>
#include<stdio.h>
#include<queue>
#include<limits>
using namespace std;

int trapRainWater(vector<vector<int> >& heightMap) {
	if(heightMap.size()==0) return 0;
	int row = heightMap.size();
	int col = heightMap[0].size();
	priority_queue<pair<int, int>, vector<pair<int, int> >, greater<pair<int, int> >
>
	que;
	vector<vector<int> >visited(row, vector<int>(col,0));
	int ans=0;int Max = INT_MIN;
	for(int i=0;i<row;i++)
	{
		for(int j=0;j<col;j++)
		{
			// (zhewei) first mark the margin of heightMap
			// do not mark the inner part
			if(!(i==0 || i==row-1 || j==0 || j==col-1)) continue;
			que.push(pair<int,int>(heightMap[i][j],i*col+j));
			visited[i][j]=1;
		}
	}
	int direct[][2] = {{0,1},{0,-1},{1,0},{-1,0}};
	while(!que.empty())
	{
		pair<int, int> tmp = que.top();que.pop();
		int height = tmp.first;
		int x = tmp.second/col; int y = tmp.second%col;
		//(zhewei) max is ordered by ascend
		Max = max(Max, height);
		for(int i=0;i<4;i++)
		{
			int x2 = direct[i][0]+x;int y2 = direct[i][1]+y;
			if(x2<0 || x2>=row || y2<0 || y2>=col || visited[x2][y2])
				continue;
			visited[x2][y2]=1;
			if(heightMap[x2][y2]<Max) ans+=Max-heightMap[x2][y2];
			que.push(make_pair<int,int>(heightMap[x2][y2],x2*col+y2));
		}
	}
	return ans;
}

int main(int argc,char *argv[])
{
	//test
	int arr[][6] = {
		{1,4,3,1,3,2},
		{3,2,1,3,2,4},
		{2,3,3,2,3,1}};
	vector<vector<int> > heightmap(3,vector<int>(6,0));
	for(int i=0;i<3;i++)
```

```
            for(int j=0;j<6;j++)
                  heightmap[i][j]=arr[i][j];
      int ans = trapRainWater(heightmap);
      cout<<ans<<endl;
      return 0;
}
```

output: 4