

木桶效应:

<https://blog.csdn.net/cll8652469346/article/details/53127801>

给定一个数组，把他们放入M*N的方格中，每个数字代表该方格的高度，这样俯视就会形成凹凸不平，如果用这个形状存储水，凹的地方会有积水，请问它能存储多少水？

例如二维数组为:

9 9 9 9

3 0 0 9

7 8 9 6

答案是中间的0 0 位置可以存储3，（因为其外面最低为3，即“木桶效应”）个单位的水。
因此答案为 $3 + 3 = 6$;

实现原理的思路展示

1. 首先，设定好方格二维数组，

2-1 其次，找到方格总的“至高点”

2-2 按照“至高点”填充所有方格，使其充满最大水量

```
final static int GRID_HEIGHT = 0; // 方格属性下标0: 方格高度
```

```
final static int GRID_WATER = 1; // 方格属性下标1: 储存水量
```

3. 然后开始移除水

3-1. 如果在最外层的边界上的话，不能存储水

3-2. 然后开始遍历整个数组，如果有能移除的水就移除掉，并且重复步骤3

直到所有的水都不能移除为止

3-3. 水是否能移除，就看当前位置的水位是否比上下左右的水位高即可（即是`gridArray[m][n][GRID_WATER] + gridArray[m][n][GRID_HEIGHT] > gridArray[m-1][n][GRID_WATER] + gridArray[m-1][n][GRID_HEIGHT]`）

4. 打印剩下的水的纵总和即可

方格高度：

9 9 9 9

3 0 0 9

7 8 9 6

加满水后，储存水量：

0 0 0 0

6 9 9 0

2 1 0 3

移除水后，储存水量：

0 0 0 0

0 9 9 0

2 1 0 3

移除水后，储存水量：

0 0 0 0

0 3 9 0

2 1 0 3

移除水后，储存水量：

0 0 0 0

0 3 3 0

2 1 0 3

移除水后，储存水量：

0 0 0 0

0 3 3 0

0 1 0 3

移除水后，储存水量：

0 0 0 0

0 3 3 0

0 0 0 3

移除水后，储存水量：

0 0 0 0

0 3 3 0

0 0 0 0

合计储存水量：6