在二维数组grid中，grid[i][j]代表位于某处的建筑物的高度。 我们被允许增加任何数量（不同建筑物的数量可能不同）的建筑物的高度。 高度 0 也被认为是建筑物。

最后，从新数组的所有四个方向（即顶部，底部，左侧和右侧）观看的“天际线”必须与原始数组的天际线相同。 城市的天际线是从远处观看时，由所有建筑物形成的矩形的外部轮廓。 请看下面的例子。

建筑物高度可以增加的最大总和是多少？

****例子：输入：**** grid = [[3,0,8,4],[2,4,5,7],[9,2,6,3],[0,3,1,0]]****输出：**** 35****解释：****

The grid is:

[ [3, 0, 8, 4],

[2, 4, 5, 7],

[9, 2, 6, 3],

[0, 3, 1, 0] ]

从数组竖直方向（即顶部，底部）看“天际线”是：[9, 4, 8, 7]

从水平水平方向（即左侧，右侧）看“天际线”是：[8, 7, 9, 3]

在不影响天际线的情况下对建筑物进行增高后，新数组如下：

gridNew = [ [8, 4, 8, 7],

[7, 4, 7, 7],

[9, 4, 8, 7],

[3, 3, 3, 3] ]

****说明:****

* 1 < grid.length = grid[0].length <= 50。
* grid[i][j] 的高度范围是： [0, 100]。
* 一座建筑物占据一个grid[i][j]：换言之，它们是 1 x 1 x grid[i][j] 的长方体。

.C#

public class Solution {

public int MaxIncreaseKeepingSkyline(int[][] grid) {

//int[][] gridnew =grid;

int i=0,j=0,i1=0,j1=0,xx=0,yy=0;

int x=grid.Length;//获取不规则数组的长

int y=grid.GetLength(0);//获取不规则数组的宽

//int[,] array = new int[4, 2];

int[,] gridnew = new int[x,y];//创建一个新的数组

for(i=0;i<x;i++){

for(j=0;j<y;j++){

gridnew[i,j]=grid[i][j];

}

}

//要判断是否本身最大

for(i=0;i<x;i++){

for(j=0;j<y;j++){

xx=0;

yy=0;

for(i1=0;i1<x;i1++){

if(xx<gridnew[i1,j]){

xx=gridnew[i1,j];

}

}

for(j1=0;j1<y;j1++){

if(gridnew[i,j1]>yy){

yy=gridnew[i,j1];

}

}

if(xx==gridnew[i,j]||yy==gridnew[i,j]){

xx=0;

yy=0;

}else{

if(xx>yy){

gridnew[i,j]=yy;

}

if(xx<yy){

gridnew[i,j]=xx;

}

if(xx==yy){

gridnew[i,j]=xx;

}

}

}

}

xx=0;

yy=0;

for(i=0;i<x;i++){

for(j=0;j<y;j++){

xx=gridnew[i,j]+xx;

yy=grid[i][j]+yy;

}

}

return xx-yy;

}

}