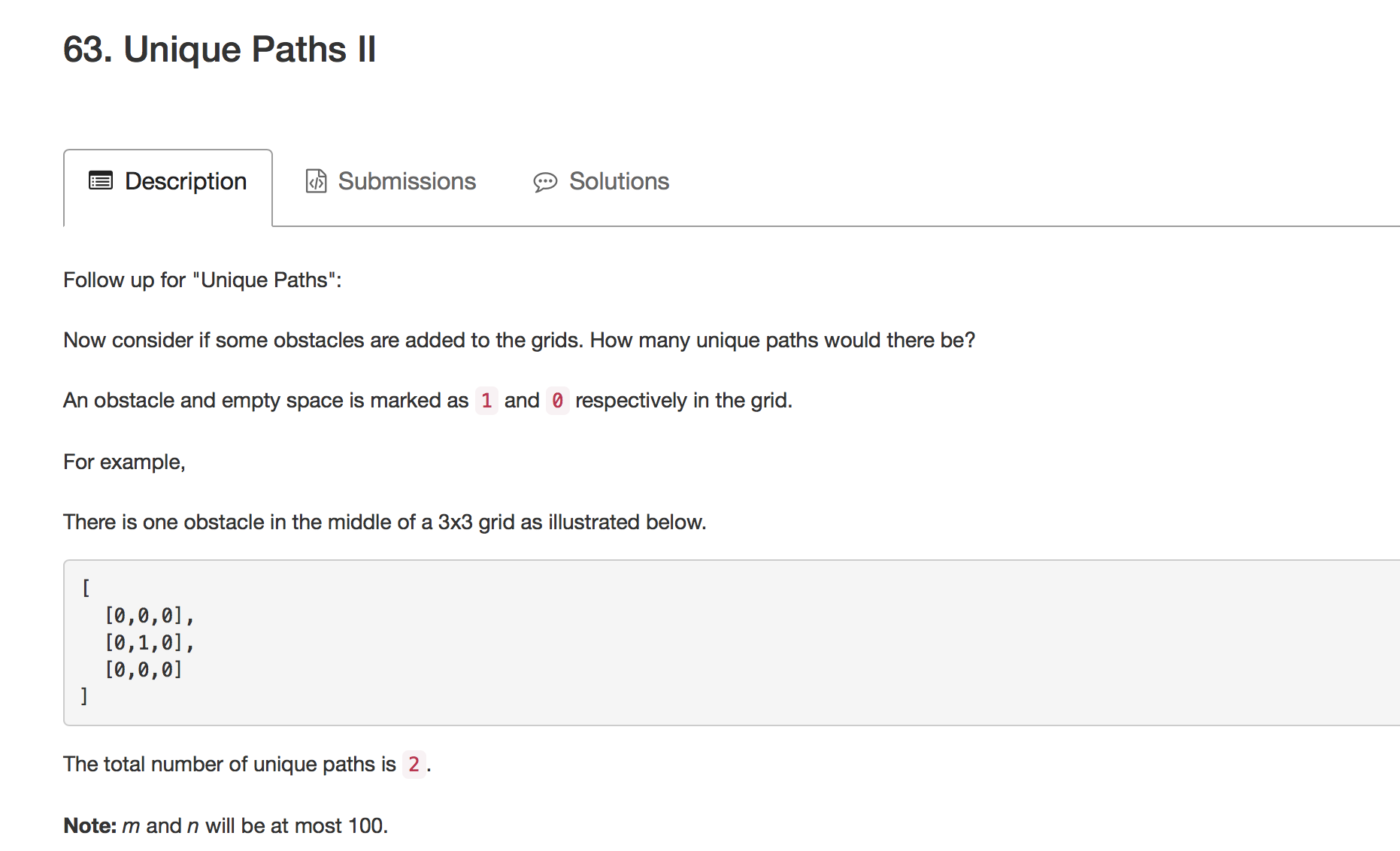
63. Unique Paths II

1.需要注意的地方已经用红色标出，就是如果遇到障碍，就清0，注意第0行和第0列特殊处理。

public class Solution {

public int uniquePathsWithObstacles(int[][] obstacleGrid) {

int m,n;

int i,j;

m=obstacleGrid.length;

n=obstacleGrid[0].length;

if(m==1)

{

for(j=0;j<n;j++)

if(obstacleGrid[m-1][j]==1) return 0;

return 1;

}

if(n==1)

{

for(i=0;i<m;i++)

if(obstacleGrid[i][n-1]==1) return 0;

return 1;

}

if((obstacleGrid[m-2][n-1]==1&&obstacleGrid[m-1][n-2]==1)||obstacleGrid[m-1][n-1]==1||obstacleGrid[0][0]==1) return 0;

for(i=1;i<m;i++)

for(j=1;j<n;j++)

{

if(obstacleGrid[i][j]==0)

obstacleGrid[i][j]=1;

else

obstacleGrid[i][j]=0;

}

for(i=0;i<m;i++)

{

if(obstacleGrid[i][0]==0)

obstacleGrid[i][0]=1;

else

break;

}

for(j=i;j<m;j++)

obstacleGrid[j][0]=0;

for(**i=1**;i<n;i++)

{

if(obstacleGrid[0][i]==0)

obstacleGrid[0][i]=1;

else

break;

}

for(j=i;j<n;j++)

obstacleGrid[0][j]=0;

for(i=1;i<m;i++)

for(j=1;j<n;j++)

{

if(obstacleGrid[i][j]==0) continue;

obstacleGrid[i][j]=obstacleGrid[i-1][j]+obstacleGrid[i][j-1];

}

return obstacleGrid[m-1][n-1];

}

}

