/\*

Given a triangle, find the minimum path sum from top to bottom. Each step you may move to adjacent numbers on the row below.

For example, given the following triangle

[

[2],

[3,4],

[6,5,7],

[4,1,8,3]

]

The minimum path sum from top to bottom is 11 (i.e., 2 + 3 + 5 + 1 = 11).

Note:

Bonus point if you are able to do this using only O(n) extra space, where n is the total number of rows in the triangle.

很明显的动态规划，当前总值只和上面的左右有关系，一次遍历就搞定

\*/

class Solution {

public:

int minimumTotal(vector<vector<int>>& tri)

{

for(int i=1;i<tri.size();i++)

{

tri[i][0]+=tri[i-1][0];

tri[i][i]+=tri[i-1][i-1];

for(int j=1;j<i;j++)

tri[i][j]+=(tri[i-1][j-1]>tri[i-1][j])?tri[i-1][j]:tri[i-1][j-1];

}

int minway=INT\_MAX;

for(int i=0;i<tri.size();i++)

minway=(minway>tri[tri.size()-1][i])?tri[tri.size()-1][i]:minway;

return minway;

}

};