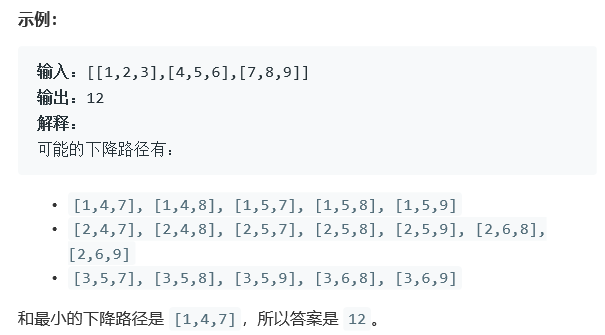
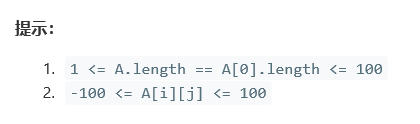
题目：

给定一个方形整数数组 A，我们想要得到通过 A 的下降路径的最小和。

下降路径可以从第一行中的任何元素开始，并从每一行中选择一个元素。在下一行选择的元素和当前行所选元素最多相隔一列。





好吧其实没有用到提示来着

本问就是一道简单的动态规划问题，同时为了减少空间消耗，我这里就只是申请了O(n)的空间，同时使用i%2的形式去完成对上下行的确认。

class Solution {

public:

int minFallingPathSum(vector<vector<int>>& A) {

int size=A.size();

if(size<2)return A[0][0];

vector<vector<int>> dp(2,vector<int>(size,0));

int s=0;

for(int i:A[0]){

dp[0][s++]=i;

}

int f1,f2;

for(int j=1;j<size;j++){

f1=j%2;

f2=1-f1;

dp[f1][0]=min(dp[f2][0],dp[f2][1])+A[j][0];

for(int i=1;i<size-1;i++){

dp[f1][i]=min(dp[f2][i],min(dp[f2][i-1],dp[f2][i+1]))+A[j][i];

}

dp[f1][size-1]=min(dp[f2][size-1],dp[f2][size-2])+A[j][size-1];

}

int res=dp[f1][0];

for(int i:dp[f1]){

res=min(res,i);

}

return res;

}

};