NOIP 2018 模拟题 day2

## 问题1 牛郎和织女（magpie.cpp）

时间限制：1s 空间限制 256M

牛郎和织女终于盼来了七夕，他们迫不及待地从自家出发，希望尽快的汇合。假设牛郎织女都在N\*M的地图上。牛郎每次移动走日字对角线，织女每次移动走田字对角线，都不能走出地图。牛郎单次移动的时间为1s，织女单次移动的时间为2s。请问牛郎和织女出发后最快花多少时间能相会？

**输入格式（magpie.in）**

输入共一行，包含两个整数，N,M。约定织女在位置（1,1），牛郎在位置（N,M）

**输出格式（magpie.out）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

输出共一行，相会的最短时间，单位s。

**样例输入**

5 6

**样例输出**

4s

**数据规模**

对于40%的数据，有N,M≤10；

对于100%的数据，有2≤N,M≤1000。

## 问题2 找同学（classmate.cpp）

时间限制：1s 空间限制 256M

小明要到他的同学家玩，可他只知道他住在某一单元，却不知住在哪个房间。那个单元有N层（1,2……N），每层有M（1,2……M）个房间。

小明会从第一层的某个房间开始找，他找的方式也很特别，每次到了一个房间，如果不是他的同学，他就会询问这人，

然后他就会去那人告诉他的房间号，如果还不是，他就会继续按这个方法找下去，直到找到。当然，他也有可能找不到同学。他的体力也是有限的，他每爬一层楼梯就会消耗V体力值。你的任务就是求找到同学耗费的最少体力值，如果找不到，就输出impossible 。

**输入格式(classmate.in)**

共N+1行。

第一行，N，M，V，X，Y（X，Y表示同学只在X层Y房间）

2~N+1行，每行M\*2个数字，第i+1行j\*2-1个数Ak和j\*2个数Ak+1表示住在i层j房间的人提供的信息是Ak层和Ak+1房间。（同学的信息就是自己的房间号）

**输出格式(classmate.out)**

输出仅一行，一个数，即答案。

**样例输入**

3 3 2 2 3

1 3 3 3 2 1

2 3 1 1 2 3

1 1 1 2 2 3

**样例输出**

2

**数据规模**

N<=1000，M<=100，V<=50

## 问题3 聪明的木匠（mj.cpp）

时间限制：1 秒 空间限制：131072 KB

一位老木匠需要将一根长的木棒切成N段。每段的长度分别为L1,L2,......,LN（1 <= L1,L2,…,LN <= 1000，且均为整数）个长度单位。我们认为切割时仅在整数点处切且没有木材损失。

木匠发现，每一次切割花费的体力与该木棒的长度成正比，不妨设切割长度为1的木棒花费1单位体力。例如：若N=3，L1 = 3,L2 = 4,L3 = 5，则木棒原长为12，木匠可以有多种切法，如：先将12切成3+9.，花费12体力，再将9切成4+5，花费9体力，一共花费21体力；还可以先将12切成4+8，花费12体力，再将8切成3+5，花费8体力，一共花费20体力。显然，后者比前者更省体力。

那么，木匠至少要花费多少体力才能完成切割任务呢？

**输入格式(mj.in)**

第1行：1个整数N(2 <= N <= 50000)

第2 - N + 1行：每行1个整数Li(1 <= Li <= 1000)。

**输出格式(mj.out)**

输出最小的体力消耗。

**样例输入**

3

3

4

5

**样例输出**

19

## 问题4 岛屿的数量（daoyu.cpp）

时间限制：1 秒 空间限制：131072 KB

有N个岛连在一起形成了一个大的岛屿，如果海平面上升超过某些岛的高度时，则这个岛会被淹没。原本的大岛屿则会分为多个小岛屿，如果海平面一直上升，则所有岛都会被淹没在水下。

给出N个岛的高度。然后有Q个查询，每个查询给出一个海平面的高度H，问当海平面高度达到H时，海上共有多少个岛屿。例如：

岛屿的高度为：{2, 1, 3, 2, 3}, 查询为：{0, 1, 3, 2}。

当海面高度为0时，所有的岛形成了1个岛屿。

当海面高度为1时，岛1会被淹没，总共有2个岛屿{2} {3, 2, 3}。

当海面高度为3时，所有岛都会被淹没，总共0个岛屿。

当海面高度为2时，岛0, 1, 3会被淹没，总共有2个岛屿{3} {3}。

**输入格式(daoyu.in)**

第1行：2个数N, Q中间用空格分隔，其中N为岛的数量，Q为查询的数量(1 <= N, Q <= 50000)。

第2 - N + 1行，每行1个数，对应N个岛屿的高度(1 <= A[i] <= 10^9)。

第N + 2 - N + Q + 1行，每行一个数，对应查询的海平面高度(1 <= Q[i] <= 10^9)。

**输出格式(daoyu.out)**

输出共Q行，对应每个查询的岛屿数量。

**样例输入**

5 4

2

1

3

2

3

0

1

3

2

**样例输出**

1

2

0

2