2018省赛前模拟R1 by DLee

【注意事项】

1. 本次比赛时长4h，难度代码量均在GDOI DAY1~DAY2。（也许吧）
2. 试题难度大概似乎是递增的。
3. **题目实现难度并不大，但是需要一定思维转化以及分步思考。**
4. 由于是网上找题+自己口胡，所以很有可能出现大家做过的情况，但是大家也要自己再打一次哦。
5. 不需要建立子文件夹，lemon读不到。
6. 测评当然是在我这台全村最烂的电脑上啦。
7. 需要输出long long或unsigned long long的题目请用windows下的输出方法(I64d,I64u)（当然因为是64位所以lld，llu也不会GG的）
8. **开启O2优化**
9. DLee不会用pdf，所以用word了。
10. 本套题目最终解释权归DLee所有。
11. 祝玩得开心。

【题目背景】

DLee很喜欢玩RPG游戏，但是这些游戏多半会遇到一些玄学的问题导致他无法完美通关。比如BUG信条，DLee很难做到在主线任务的限制区域不被发现。因此DLee找到了你，想让你帮助他设计通关方式。

【题目信息】

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 试题名称 | 黎明杀机 | 孤岛惊魂 | 使命召唤 | 古墓丽影 |
| 题目文件名 | daylight | cry | duty | tomb |
| 输入文件名 | daylight.in | cry.in | duty.in | tomb.in |
| 输出文件名 | daylight.out | cry.out | duty.out | tomb.out |
| 时间限制 | 1s | 1s | 2.5s | 1s |
| 空间限制 | 128MB | 128MB | 512MB | 128MB |
| 测试点数目 | 10 | 10 | 10 | 10 |
| 单点分值 | 10 | 10 | 10 | 10 |
| 评测环境 | Windows | Windows | Windows | Windows |

【T1】黎明杀机（daylight）

【题目描述】

DLee今天在玩黎明杀机（Dead by Daylight）。黎明杀机里，屠夫需要将人类砍倒并挂在钩子上以获得分数，而人类需要通过修发电机打开大门或者等待长久的时间用钥匙打开地道逃生。

**黎明杀机的地图可以看作是一幅n个点m条边的无向图**，屠夫想尽可能得分，要做的就是用自己的技能和风骚的走位来限制人类逃生。

DLee作为一名屠皇，各个屠夫都被他玩得十分娴熟，以至于基本上每次游戏都能挂死三个人类或屠场（一共四个人类）。**不过黎明杀机随机出的地图不一定联通，这也使得DLee可能会少挂几个人。**

今天黎明杀机出了一个新的屠夫，他的技能是修改地图。**更简单地说，他每次可以选择两个点u和v，将他们合并成一个点w。接着，对于，原图中的一条边(u,i)或(v,i)，连边(w,i)，对于边(u,u)或(v,v)或(u,v)，连边(w,w)。这样每次使用技能后地图点数减少1，边数不变。**

DLee想利用这个技能更方便快捷地挂人，所以**他想把这幅图修改成不存在重边和简单环，但可以存在自环的图。同时这幅图应该要存在一条路径使得所有不在路径上的点到这条路径的距离为1。**这样DLee就可以从一个点出发一路杀过去挂人了。

现在DLee用这个屠夫不够娴熟，因此他想编个程序看看自己**最少要用多少次技能才能使地图满足条件**。但是因为他一直在打游戏，所以他将这个任务交给了你。

【输入格式】

第一行两个整数n和m，表示无向图大小。

接下来m行，每行两个整数u和v，表示u,v之间有一条无向边连接。

【输出格式】

一行仅一个整数，表示DLee最少使用技能次数。

【输入样例】

输入1：

4 4

1 2

2 3

3 4

4 2

输入2：

6 3

1 2

3 4

5 6

输入3：

7 6

1 2

2 3

1 4

4 5

1 6

6 7

【输出样例】

输出1：

2

输出2：

2

输出3：

1

【样例解释】

应该都看得懂，不解释了。

【数据范围】

对于20%的数据，n,m<=10

对于另20%的数据，n，m<=2000

对于另20%的数据，给出的图是一颗树

对于全部数据，1<=n,m<=10^5。

【T2】孤岛惊魂（cry）

【题目描述】

DLee今天在玩孤岛惊魂（Far Cry）。孤岛惊魂里，有一个邪教组织与我们作对，我们需要通过一系列线索找到他们的据点并摧毁。当然在这个过程中我们还要注意那些丧尸小怪，有时他们也会带来危险。

以DLee的聪明才智，找到邪教组织的据点自然不在话下，但是有一种东西叫做剧情杀，所以经常踩flag的DLee感觉很郁闷。

今天DLee在打一个据点的时候，遇到了一扇机关门，**门的上面是一排长度为n的固定的能量宝石，每颗宝石的能量为ai，而DLee手上有m颗可以随意摆放位置的能量宝石，每颗宝石的能量为bi。这扇门的能量需求为h，宝石需求是m。**

DLee已经玩了很久了，自然知道应该怎么打开：**只需要在这些固定的能量宝石中选择连续的m个，并且在每一个下面放上一颗宝石，使得上下两颗能量宝石的能量和不小于h就行了。**

DLee轻而易举地就解决了这个问题，但是他又遭遇了剧情杀。现在郁闷的DLee又来到了这扇门前，现在他想知道，用他手上的宝石，一共有多少种不同的方案使得门能被打开。

但是DLee想了想，这个问题太难了，于是他简化了一下问题，他现在想知道，**他能够选择固定宝石中的多少种连续子序列（显然可能的选择方案最多为n-m+1），使得他用手上的宝石能够打开门。**

DLee怕你们还是不清楚，所以他给出了样例解释。

【输入格式】

第一行三个整数n,m,h

第二行有m个数字，为bi

第三行有n个数字，为ai。

【输出格式】

一行仅一个整数，表示DLee能够选择的连续子序列个数。

【输入样例】

输入1：

5 2 10

5 3

1 8 5 5 7

输入2：

4 2 10

5 5

9 3 8 9

【输出样例】

输出1：

2

输出2：

1

【样例解释】

对于样例1，DLee可以选择的区间是[2,3]和[4,5]，对于[2,3]我们让b中的5和a中的5配对，b中的3和a中的8配对即可。

对于样例2，DLee可以选择的区间是[3,4]。

【数据范围】

对于10%的数据1<=n<=10

对于40%的数据，1<=n<=1000

对于除上述外另10%的数据，ai>h

对于100%的数据，1<=n<=200000，m<=n，0<=ai,bi,h<=10^9

【T3】使命召唤（duty）

【题目描述】

DLee今天在玩使命召唤（Call of Duty）。使命召唤是一款十分用心的FPS，不仅游戏引擎让人很舒服，而且在作画贴图方面也十分细致。DLee作为剧情向玩家，在玩使命召唤时，更是感受到了剧情设计的精巧。

使命召唤的关卡类型多种多样，有潜入敌方阵营的，正面冲锋的，也有侧面支援的。而今天DLee打到的关卡是控制“上帝”导弹装置，对敌方地面阵地进行打击。

“上帝”导弹装置的控制十分特殊，**它的面板是一个可以自由扩大的网格矩形。每次打击需要在网格四条边上的格点处分别选择一个点（不包括角上），这次打击的范围就是这四个点顺次连接的范围。**

但是因为种种原因，我们选择的范围必须是一个平行四边形，这大大限制了DLee的操作空间，不过并不能阻止他通关。但是DLee是个精益求精的人，因此他不想浪费打击范围。**我们将单位网格的边长看作1，这样每种打击都有一个打击面积。现在DLee想计算出对于每一个打击面积会有多少种打击方式。注意，因为敌人来袭方向是不同的，所以如果将某一种打击方式旋转后(90,180,270)，这两种方式打击范围不完全相同，那么这两种视为不同方式。（详见样例解释）**

DLee并不需要你输出每一种打击面积有多少种打击方式，**但是他会有多个询问，每次询问打击面积在一个区间内时，哪种面积的打击方式最多，并且你要告诉DLee打击方式最多的打击面积是什么。（详见样例解释）**

【输入格式】

第一行一个整数n，表示询问次数。

接下来n行，每行两个整数sl，sr表示询问打击面积区间[sl,sr]。

【输出格式】

对于每个询问，输出一行两个整数，s和x，表示打击面积为s的打击方式最多，有x种。如果有多个不同的s打击方式一样多，输出最小的一个s。

【输入样例】

输入1：

1

4 4

输入2：

1

2 6

【输出样例】

输出1：

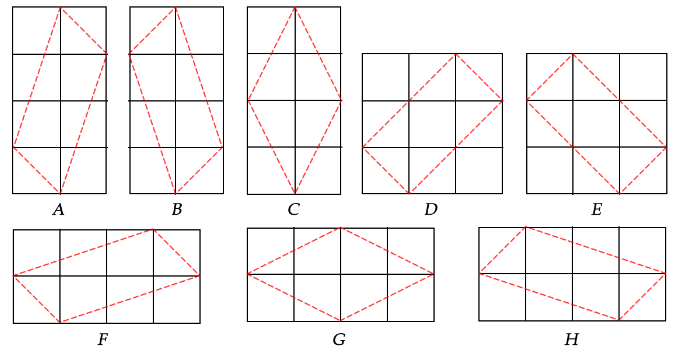
4 8

输出2：

6 20

【样例解释】

对于样例1，面积为4的可以这样搞。显然面积是其他的都没有面积为4的多。



【数据范围】

对于20%的数据，n<=10，sl,sr<=200

对于30%的数据，n<=100000，sl，sr<=200

对于60%的数据，n<=100000，sl，sr<=5000

对于100%的数据，n<=100000，sl<=sr<=2\*10^5

【T4】古墓丽影（tomb）

【题目描述】

DLee今天在玩古墓丽影（Tomb Raider）。古墓丽影是一款操纵着性感的劳拉在山林雪原古墓间，探寻真相，顺手盗墓的游戏。游戏时你会遇到为了金钱或是某些神力前来盗墓的有钱人，会遇到一些木乃伊之类的玄学现象，当然也少不了机关解密这类东西。

DLee今天在玩耍的时候又遇到了一个弱鸡谜题，谜题是这样的：要想进入古墓，需要将古墓的钥匙调整到符合古墓锁孔的形态。古墓的锁孔很奇特，它可以看做一棵由n个节点组成的无根树，每个节点上有一个标记0或1。现在你的钥匙是一棵和锁孔长得一样的东西，只是节点上的标记不一样了。**DLee想让你告诉他，他最少需要将多少个节点上的进行标记修改。**

不，等等！DLee发现，钥匙的每个分支都是可以旋转的！**也就是说，钥匙不需要调整到和锁孔一模一样，只需要让钥匙在重标号后与锁孔同构就可以了！**

**同构的意思是，存在一个1到n的置换P，是的对于锁孔任意节点之间的边u-v，都一定有钥匙节点之间的边P(u)-P(v)，同时如果不存在锁孔节点的边u-v，也一定不存在钥匙节点的边P(u)-P(v)。**

但是这个问题并不简单，DLee想了很久也想不出来，所以他想让你帮助他计算出这种情况下的最小修改次数，

【输入格式】

第一行一个整数n，表示锁孔的大小，节点标号为1~n。

接下来n-1行，每行2个整数u和v，表示这两个节点之间有一条无向边。

接下来一行有n个整数，表示锁孔的形态。

最后一行有n个整数，表示钥匙的当前形态。

【输出格式】

一行仅一个整数，表示DLee的最小修改次数。

【输入样例】

输入1：

4

1 2

2 3

3 4

0 0 1 1

1 0 0 0

【输出样例】

1

【样例解释】

只需要修改钥匙的2号节点，这样重编号后与锁孔同构。

【数据范围】

对于30%的数据，n<=10。

对于60%的数据，n<=100。

对于另10%的数据，锁孔的形态是一条链。

对于100%的数据，n<=700，每个点的度数不超过11。