CCF 全国信息学奥林匹克联赛真题模拟赛

提高组 day1

（请选手务必仔细阅读本页内容）

一．题目概况

中文题目名称	货币系统	飞扬的小鸟	宝藏	疫情控制
英文题目与子目录名	money	bird	treasure	blockade
可执行文件名	money	bird	treasure	blockade
输入文件名	money.in	bird.in	treasure.in	blockade
输出文件名	money.out	bird.out	treasure.out	blockade
每个测试点时限	1 秒	1 秒	1 秒	2 秒
测试点数目	20	20	20	10
每个测试点分值	5	5	5	10
附加样例文件	有	有	有	有
结果比较方式	全文比较（过滤行末空格及文末回车）			
题目类型	传统	传统	传统	传统
运行内存上限	512M	128M	256M	128M

注意事项：

1、文件名（程序名和输入输出文件名）必须使用英文小写。

2、C/C++中函数 main()的返回值类型必须是 int，程序正常结束时的返回值必须是 0。

3、全国统一评测时采用的机器配置为：CPU AMD Athlon(tm) II x2 240 processor，2.8GHz，

内存 4G，上述时限以此配置为准。

4、只提供 Linux 格式附加样例文件。

5、提交的程序代码文件的放置位置请参照各省的具体要求。

6、特别提醒：评测在当前最新公布的 **NOI Linux** 下进行，各语言的编译器版本以其为准。

1．货币系统

(money.cpp/c/pas)

【问题描述】

在网友的国度中共有 n 种不同面额的货币，第 i 种货币的面额为 a[i]，你可以假设每一种货币都有无穷多张。为了方便，我们把货币种数为 n、面额数组为 a[1..n] 的货币系统记作 (n,a)。

在一个完善的货币系统中，每一个非负整数的金额 x 都应该可以被表示出，即对每一个非负整数 x，都存在 n 个非负整数 t[i] 满足 a[i]×t[i] 的和为 x。然而，在网友的国度中，**货币系统可能是不完善的**，即可能存在金额 x 不能被该货币系统表示出。例如在货币系统 n=3, a=[2,5,9] 中，金额 1,3 就无法被表示出来。

两个货币系统 (n,a) 和 (m,b) 是等价的，当且仅当**对于任意非负整数 x，它要么均可以被两个货币系统表出，要么不能被其中任何一个表出。**

现在网友们打算简化一下货币系统。他们希望找到一个货币系统 (m,b)，满足 (m,b) 与原来的货币系统 (n,a) 等价，且 m 尽可能的小。他们希望你来协助完成这个艰巨的任务：找到最小的 m。

【输入格式】

输入文件名为 money.in。

输入文件的第一行包含一个整数 T，表示数据的组数。接下来按照如下格式分别给出 T 组数据。

每组数据的第一行包含一个正整数 n。接下来一行包含 n 个由空格隔开的正整数 a[i]。

【输出格式】

输出文件名为 money.out。

输出文件共有 T 行，对于每组数据，输出一行一个正整数，表示所有与 (n,a) 等价的货币系统 (m,b) 中，最小的 m。

【输入输出样例 1】

money.in	**money.out**
2 4 3 19 10 6 5 11 29 13 19 17	2 5

见选手目录下的 money/money1.in 和 money/money1.ans。

【输入输出样例 1 说明】

在第一组数据中，货币系统 (2, [3,10]) 和给出的货币系统 (n, a) 等价，并可以验证不存在 m < 2 的等价的货币系统，因此答案为 2。

在第二组数据中，可以验证不存在 m < n 的等价的货币系统，因此答案为 5。

【输入输出样例 2】

见选手目录下的 money/money2.in 和 money/money2.ans。

【数据规模与约定】

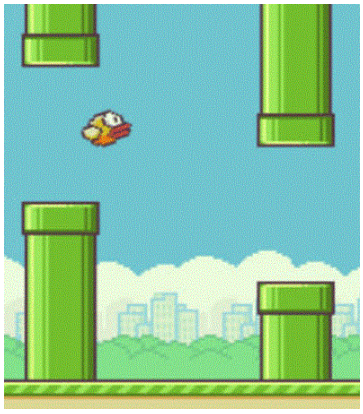
测试点	n	a_i	测试点	n	a_i
1	$=2$	≤ 1000	11	≤ 13	≤ 16
2			12		
3			13		
4	$=3$		14	≤ 25	≤ 40
5			15		
6			16		
7	$=4$		17	≤ 100	≤ 25000
8			18		
9	$=5$		19		
10			20		

对于 100% 的数据，满足 1 ≤ T ≤ 20, n,a[i] ≥ 1。

2. 飞扬的小鸟

(bird.cpp/c/pas)

【问题描述】



Flappy Bird 是一款风靡一时的休闲手机游戏。玩家需要不断控制点击手机屏幕的频率来调节小鸟的飞行高度，让小鸟顺利通过画面右方的管道缝隙。如果小鸟一不小心撞到了水管或者掉在地上的话，便宣告失败。

为了简化问题，我们对游戏规则进行了简化和改编：

1. 游戏界面是一个长为 n，高为 m 的二维平面，其中有 k 个管道（忽略管道的宽度）。
2. 小鸟始终在游戏界面内移动。小鸟从游戏界面最左边任意整数高度位置出发，到达游戏界面最右边时，游戏完成。
3. 小鸟每个单位时间沿横坐标方向右移的距离为 1，竖直移动的距离由玩家控制。如果点击屏幕，小鸟就会上升一定高度 X，每个单位时间可以点击多次，效果叠加；如果不点击屏幕，小鸟就会下降一定高度 Y。小鸟位于横坐标方向不同位置时，上升的高度 X 和下降的高度 Y 可能互不相同。
4. 小鸟高度等于 0 或者小鸟碰到管道时，游戏失败。小鸟高度为 m 时，无法再上升。

现在，请你判断是否可以完成游戏。如果可以，输出最少点击屏幕数；否则，输出小鸟最多可以通过多少个管道缝隙。

【输入】

输入文件名为 bird.in。

第 1 行有 3 个整数 n，m，k，分别表示游戏界面的长度，高度和水管的数量，每两个整数之间用一个空格隔开；

接下来的 n 行，每行 2 个用一个空格隔开的整数 X 和 Y，依次表示在横坐标位置 0~$n-1$ 上玩家点击屏幕后，小鸟在下一位置上升的高度 X，以及在这个位置上玩家不点击屏幕时，小鸟在下一位置下降的高度 Y。

接下来 k 行，每行 3 个整数 P，L，H，每两个整数之间用一个空格隔开。每行表示一个管道，其中 P 表示管道的横坐标，L 表示此管道缝隙的下边沿高度为 L，H 表示管道缝隙上边沿的高度（输入数据保证 P 各不相同，但不保证按照大小顺序给出）。

【输出】

输出文件名为 bird.out。

共两行。

第一行，包含一个整数，如果可以成功完成游戏，则输出 1，否则输出 0。

第二行，包含一个整数，如果第一行为 1，则输出成功完成游戏需要最少点击屏幕数，否则，输出小鸟最多可以通过多少个管道缝隙。

【输入输出样例 1】

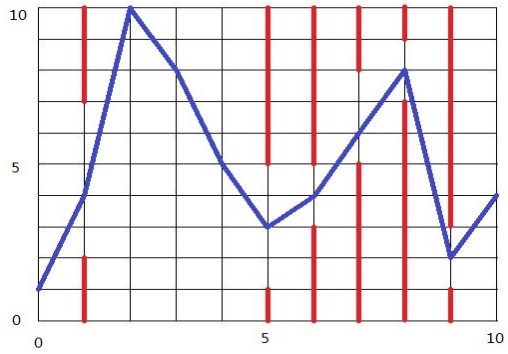
bird.in	bird.out
10 10 6 3 9 9 9 1 2 1 3 1 2 1 1 2 1 2 1 1 6 2 2 1 2 7 5 1 5 6 3 5 7 5 8 8 7 9 9 1 3	1 6

【输入输出样例 2】

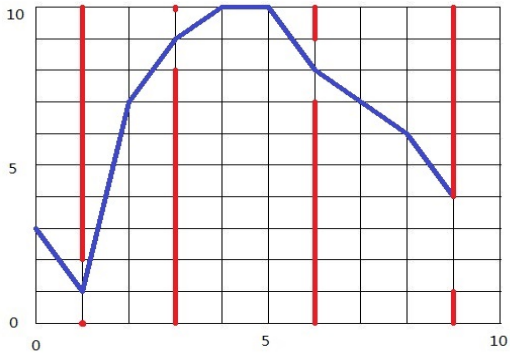
bird.in	bird.out
10 10 4 1 2 3 1 2 2 1 8 1 8 3 2 2 1 2 1 2 2 1 2 1 0 2 6 7 9 9 1 4 3 8 10	0 3

【输入输出样例说明】

如下图所示，蓝色直线表示小鸟的飞行轨迹，红色直线表示管道。



输入输出样例1说明



输入输出样例2说明

【数据范围】

对于 30%的数据：$5\leq n\leq 10$，$5\leq m\leq 10$，$k=0$，保证存在一组最优解使得同一单位时间最多点击屏幕 3 次；

对于 50%的数据：$5\leq n\leq 20$，$5\leq m\leq 10$，保证存在一组最优解使得同一单位时间最多点击屏幕 3 次；

对于 70%的数据：$5\leq n\leq 1000$，$5\leq m\leq 100$；

对于 100%的数据：$5\leq n\leq 10000$，$5\leq m\leq 1000$，$0\leq k<n$，$0<X<m$，$0<Y<m$，$0<P<n$，$0\leq L<H\leq m$，$L+1<H$。

3. 宝藏

（treasure.cpp/c/pas）

【问题描述】

参与考古挖掘的小明得到了一份藏宝图，藏宝图上标出了 n 个深埋在地下的宝藏屋，也给出了这 n 个宝藏屋之间可供开发的 m 条道路和它们的长度。

小明决心亲自前往挖掘所有宝藏屋中的宝藏。但是，每个宝藏屋距离地面都很远，也就是说，从地面打通一条到某个宝藏屋的道路是很困难的，而开发宝藏屋之间的道路则相对容易很多。

小明的决心感动了考古挖掘的赞助商，赞助商决定免费赞助他打通一条从地面到某个宝藏屋的通道，通往哪个宝藏屋则由小明来决定。

在此基础上，小明还需要考虑如何开凿宝藏屋之间的道路。已经开凿出的道路可以任意通行不消耗代价。每开凿出一条新道路，小明就会与考古队一起挖掘出由该条道路所能到达的宝藏屋的宝藏。另外，小明不想开发无用道路，即两个已经被挖掘过的宝藏屋之间的道路无需再开发。

新开发一条道路的代价是：

这条道路的长度 × 从赞助商帮你打通的宝藏屋到这条道路起点的宝藏屋所经过的宝藏屋的数量（包括赞助商帮你打通的宝藏屋和这条道路起点的宝藏屋）。

请你编写程序为小明选定由赞助商打通的宝藏屋和之后开凿的道路，使得工程总代价最小，并输出这个最小值。

【输入格式】

输入文件名为 treasure.in。

第一行两个用空格分离的正整数 n 和 m，代表宝藏屋的个数和道路数。

接下来 m 行，每行三个用空格分离的正整数，分别是由一条道路连接的两个宝藏屋的编号（编号为 1~n），和这条道路的长度 v。

【输出格式】

输出文件名为 treasure.out。

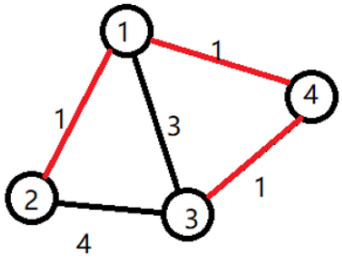
输出共一行，一个正整数，表示最小的总代价。

【输入输出样例 1】

treasure.in	treasure.out
4 5 1 2 1 1 3 3 1 4 1 2 3 4 3 4 1	4

见选手目录下的 treasure/treasure1.in 与 treasure/treasure1.ans

【输入输出样例 1 说明】



小明选定让赞助商打通了 1 号宝藏屋。小明开发了道路 1→2，挖掘了 2 号宝藏。开发了道路 1→4，挖掘了 4 号宝藏。还开发了道路 4→3，挖掘了 3 号宝藏。工程总代价为：1×1 + 1×1 + 1×2 =4

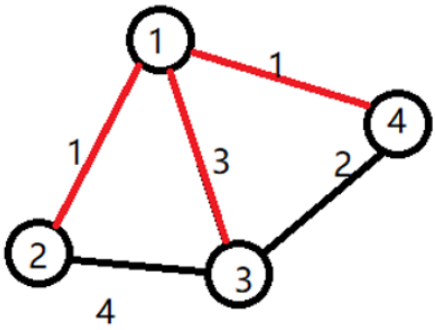
(1→2) (1→4) (4→3)

【样例输入输出 2】

treasure.in	**treasure.out**
4 5 1 2 1 1 3 3 1 4 1 2 3 4 3 4 2	5

见选手目录下的 treasure/treasure2.in 与 treasure/treasure2.ans。

【输入输出样例 2 说明】



小明选定让赞助商打通了 1 号宝藏屋。小明开发了道路 1→2，挖掘了 2 号宝藏。开发了道路 1→3，挖掘了 3 号宝藏。还开发了道路 1→4，挖掘了 4 号宝藏。工程总代价为：1×1 + 3×1 + 1×1 =5

(1→2) (1→3) (1→4)

【输入输出样例 3】

见选手目录下的 treasure/treasure3.in 和 treasure/treasure3.out。

【数据规模与约定】

对于 20%的数据：

保证输入是一棵树，1≤n≤8，v≤5000 且所有的 v 都相等。

对于 40%的数据：

1≤n≤8，0≤m≤1000，v≤5000 且所有的 v 都相等。

对于 70%的数据：

1≤n≤8，0≤m≤1000，v≤ 5000

对于 100%的数据：

1≤n≤12，0≤m≤1000，v≤ 500000

4. 疫情控制

（**blockade.cpp/c/pas**）

【问题描述】

H 国有 n 个城市，这 n 个城市用 n-1 条双向道路相互连通构成一棵树，1 号城市是首都，也是树中的根节点。

H 国的首都爆发了一种危害性极高的传染病。当局为了控制疫情，不让疫情扩散到边境城市（叶子节点所表示的城市），决定动用军队在一些城市建立检查点，使得从首都到边境城市的每一条路径上都至少有一个检查点，边境城市也可以建立检查点。但特别要注意的是，首都是不能建立检查点的。

现在，在 H 国的一些城市中已经驻扎有军队，且一个城市可以驻扎多个军队。一支军队可以在有道路连接的城市间移动，并在除首都以外的任意一个城市建立检查点，且只能在一个城市建立检查点。一支军队经过一条道路从一个城市移动到另一个城市所需要的时间等于道路的长度（单位：小时）。

请问最少需要多少个小时才能控制疫情。注意：不同的军队可以同时移动。

【输入】

输入文件名为 blockade.in。

第一行一个整数 n，表示城市个数。

接下来的 n-1 行，每行 3 个整数，u、v、w，每两个整数之间用一个空格隔开，表示从城市 u 到城市 v 有一条长为 w 的道路。数据保证输入的是一棵树，且根节点编号为 1。

接下来一行一个整数 m，表示军队个数。

接下来一行 m 个整数，每两个整数之间用一个空格隔开，分别表示这 m 个军队所驻扎的城市的编号。

【输出】

输出文件为 blockade.out。

共一行，包含一个整数，表示控制疫情所需要的最少时间。如果无法控制疫情则输出-1。

【输入输出样例】

blockade.in	blockade.out
4 1 2 1 1 3 2 3 4 3 2 2 2	3

【输入输出样例说明】

第一支军队在 2 号点设立检查点，第二支军队从 2 号点移动到 3 号点设立检查点，所需时间为 3 个小时。

【数据范围】

保证军队不会驻扎在首都。

对于 20%的数据，$2 \le n \le 10$；

对于 40%的数据，$2 \le n \le 50$，$0 < w < 10^5$；

对于 60%的数据，$2 \le n \le 1000$，$0 < w < 10^6$；

对于 80%的数据，$2 \le n \le 10{,}000$； 对于

100%的数据，$2 \le m \le n \le 50{,}000$，$0 < w < 10^9$。