

全国信息学奥林匹克联赛（NOIP2017）复赛模拟

提高组第一试

2017 年 11 月 4 日 8:30-12:00

（请选手务必仔细阅读本页内容）

题目名称	抢气球	轰炸	玩游戏
题目类型	传统	传统	传统
目录	balloon	bomb	game
可执行文件名	balloon	bomb	game
输入文件名	balloon.in	bomb.in	game.in
输出文件名	balloon.out	bomb.out	game.out
每个测试点时限	1 秒	1 秒	3 秒
内存限制	512M	512M	512M
测试点数目	20	20	20
每个测试点分值	5	5	5

提交源程序文件名

对于 C++语言	balloon.cpp	bomb.cpp	game.cpp
对于 C 语言	balloon.c	bomb.c	game.c
对于 pascal 语言	balloon.pas	bomb.pas	game.pas

编译选项

对于 C++语言	-lm	-lm	-lm
对于 C 语言	-lm	-lm	-lm
对于 pascal 语言			

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. 除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
3. 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。
4. 全国统一评测时采用的机器配置为：CPU 2.8GHz，内存 4G，上述时限以此配置为准。
5. 只提供 Linux 格式附加样例文件。
6. 评测在 NOI Linux 下进行。
7. 编译时不打开任何优化选项。
8. 最终测试时，打开-O2 优化

1. 抢气球

(balloon)

【问题描述】

小A教室的墙上挂满了气球，五颜六色，小朋友们非常喜欢。

刚一下课，小朋友们就打算去抢这些气球。每个气球在墙上都有一定的高度，只有当小朋友跳起来时，手能够到的高度大于等于气球的高度，小朋友才能摘到这个气球。为了公平起见，老师让跳的低的小朋友先摘，跳的高的小朋友后摘。小朋友都很贪心，每个小朋友在摘气球的时候都会把自己能摘的气球都摘掉。

很巧的是，小朋友们跳起来手能够着的高度都不一样，这样就不会有跳起来后高度相同的小朋友之间发生争执了。

【输入格式】

第一行输入两个空格分隔的整数 n, m ，其中 n 表示小朋友的数量， m 表示墙上气球的数量。

第二行输入 n 个正整数（每两个整数之间用空格隔开），第 i 个数为 a_i ，表示第 i 个小朋友跳起来手能够着的高度为 a_i 。

第三行输入 m 个正整数（每两个整数之间用空格隔开），第 i 个数为 h_i ，表示第 i 个气球的高度为 h_i 。

【输出格式】

输出一共 n 行，每行一个整数。

第 i 行表示第 i 个小朋友摘到的气球数量。

【输入输出样例1】

balloon.in	balloon.out
5 6	3
3 7 9 6 4	0
1 2 3 4 5 6	0
	2
	1

【样例1解释】

对于第一组样例输入，摘取气球的顺序依次为1,5,4,2,3号小朋友。1号小朋友能摘1,2,3号气球，5号小朋友能摘4号气球，4号小朋友能摘5,6号气球，2,3号小朋友没有气球可摘了。

【输入输出样例2】

balloon.in	balloon.out
10 10	1
1 2 3 4 5 6 7 8 9 10	0
3 1 4 6 7 8 9 9 4 12	1
	2
	0
	1
	1
	1
	2
	0

【数据规模与约定】

测试点	n	m	ai	hi	特殊性质
1-4	≤ 10	≤ 10	≤ 100	≤ 100	无
5-8	≤ 1000	≤ 1000	≤ 100000	≤ 100000	输入的ai依次递增
9-12	≤ 1000	≤ 1000	≤ 100000	≤ 100000	无
13-14	≤ 100000	≤ 100000	≤ 100000	≤ 100000	所有气球的高度一样
15-16	≤ 100000	≤ 100000	≤ 100000	≤ 100000	无
17-20	≤ 100000	≤ 100000	$\leq 10^9$	$\leq 10^9$	无

2. 轰炸

(bomb)

【题目描述】

战狂也在玩《魔方王国》。他只会征兵而不会建城市，因此他决定对小奇的城市进行轰炸。

小奇有 n 座城市，城市之间建立了 m 条有向的地下通道。战狂会发起若干轮轰炸，每轮可以轰炸任意多个城市。

每座城市里都有战狂部署的间谍，在城市遭遇轰炸时，它们会通过地下通道撤离至其它城市。非常不幸的是，在地道里无法得知其它城市是否被轰炸，如果存在两个不同的城市 i, j ，它们在同一轮被轰炸，并且可以通过地道从城市 i 到达城市 j ，那么城市 i 的间谍可能因为撤离到城市 j 而被炸死。为了避免这一情况，战狂不会在同一轮轰炸城市 i 和城市 j 。

你要求出战狂最少需要多少轮可以对每座城市都进行至少一次轰炸

【输入数据】

第一行两个整数 n, m 。接下来 m 行每行两个整数 a, b 表示一条从 a 连向 b 的单向边。

【输出数据】

输出一行仅一个整数表示答案。

【样例输入输出】

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

【数据范围】

对于 20% 的数据， $n, m \leq 10$ 。

对于 40% 的数据， $n, m \leq 1000$ 。

对于另外 30% 的数据，保证无环。

对于 100% 的数据， $n, m \leq 1000000$ 。

3. 玩游戏

(game)

【问题描述】

小 A 得了忧郁综合症，小 B 正在想办法开导她。

机智的小 B 决定陪着小 A 玩游戏，他从魔法的世界里变出一张无向联通图，每条边上都有边权。小 B 定义一条路径的权值为所有经过边中的**最大权值**，小 A 则定义两点的最短路径为所有路径中**权值最小**的路径权。

每次小 A 先选出两个点 $m1, m2$ ，然后小 B 选出两个点 $b1, b2$ ，计算出它们的最短路径 m, b ，然后小 B 会拿出两堆灵魂宝石，一堆有 m 个，另一堆有 b 个。然后小 A 先从一堆中选出若干个灵魂宝石拿走，接下来小 B 重复同样的操作，如此反复，直到取走最后一颗灵魂宝石，然后取走最后一颗宝石的人获胜。

小 B 认为这样游戏太简单，于是他会不定期向这张图上加上一些边，以增大游戏难度。

小 A 具有预知未来的能力，她看到了自己和小 B 在未来游戏中的选择，以及小 B 增加的边。现在对于每次游戏，小 A 想知道自己是否存在必胜的方法。但是预知未来已经消耗了她太多精力，出于疲惫她只好找到了你。

【输入格式】

第一行两个数 N 和 M ，表示这张无向图初始的点数与边数；

接下来 M 行，每行三个数 u, v, q ，表示点 u 和点 v 之间存在一条权值为 q 的边；

接下来一行一个数 Q ，表示操作总数；

接下来 Q 行，表示操作，每行格式为下面两条中的一条：

1.add $u\ v\ q$ ：表示在 u 与 v 之间加上一条边权为 q 的边；

2.game $m1\ m2\ b1\ b2$ ：表示一次游戏，其中馒头卡的选择点 $m1, m2$ ，包子卡的选择点 $b1, b2$ 。

数据保证 $1 \leq u, v, m1, m2, b1, b2 \leq n$ ， $1 \leq q$ ， $m1 \neq m2$ 且 $b1 \neq b2$

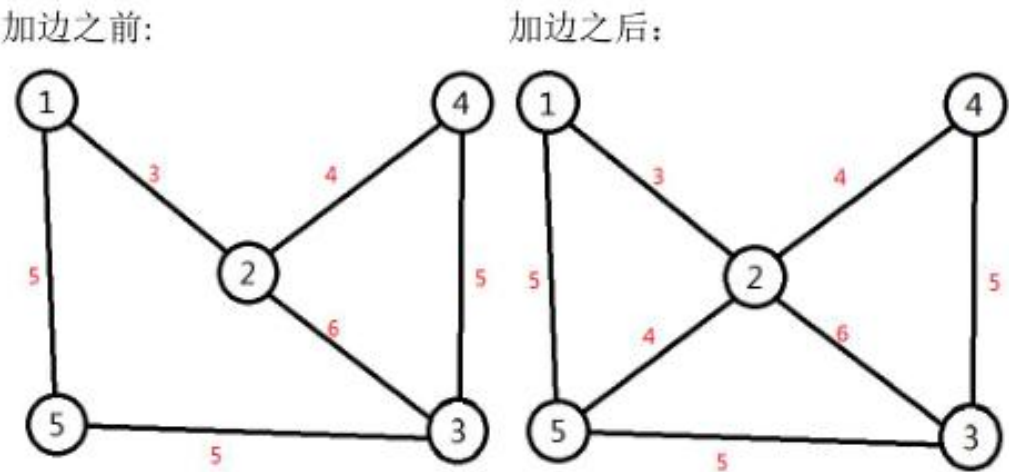
【输出格式】

对于每个 game 输出一行，若小 A 存在必胜策略，则输出“madoka”，否则输出“Baozika”，以回车结尾

【样例输入输出】

game.in	game.out
5 6	Baozika
1 2 3	madoka
2 3 6	madoka
4 2 4	
5 3 5	
3 4 5	
5 1 5	
4	
game 1 3 4 3	
game 1 5 2 4	
add 2 5 4	
game 1 5 3 4	

【样例解释】



对于第一轮，1 至 3 的最短路径为 5，4 至 3 的最短路径为 5
对于第二轮，1 至 5 的最短路径为 5，2 至 4 的最短路径为 4

再看加边之后：
对于第三轮，1 至 5 的最短路径为 4，3 至 4 的最短路径为 5

【数据规模与约定】

测试点编号	N	M	Q	边权 q	备注			
1	15	20	100	≤10000	无			
2								
3								
4								
5	1000	3000	3000					
6								
7								
8								
9								
10								
11	1000	100000	100000	998244353	add 操作数≤5000			
12				16007				
13				2000	100000	100000	≤10 ¹⁵	所有 add 操作在 game 操作之前，add 操作 ≤1000
14								
15								
16								
17								
18								
19	5000	100000	150000					
20								