

十一测试赛

Shangkun Shen

October 6, 2014

比赛日期	Oct 7th, 2014
赛制	OI 赛制
比赛时间	5 小时
题目数量	5 题
分值设定	每题 100 分, 满分 500 分

- 本次测试比赛采用 OI 赛制, 参赛选手可以使用 Pascal, C/C++, Java 来完成比赛题目。最后比赛总分得分最高者即为冠军, 其他名次依得分顺延。
- 所有的数据读入读出采用文件读入读出形式, 即要求必须使用文件输入输出, 并且文件名必须符合题目要求, 否则将不会得到成绩。
- 使用 Java 的同学, 主类名为 Main, 代码的文件名根据题目中的说明来正确命名即可。
- 题目的时间限制已经按照 2 倍标程 (C/C++) 进行了放宽处理, 而 Java 按照 3 倍时限进行放宽处理。

A. 丹丹的生日派对

时间限制	1s
空间限制	64MB
源程序名	queue.{c,cpp,java}
输入文件	queue.in
输出文件	queue.out

问题描述

丹丹的生日就快到了，她的好伙伴们一直在帮她策划一个新的活动，没错，就是举办一个派对！这样她的好朋友们都可以过来一起愉快的玩耍，并且能够分享她的喜悦了！

但是令丹丹意外的是，来的人实在是太多了，太多了！多得让丹丹的头都大了。在这群排队的人中，有男生，有女生，还有熊孩子（既然叫熊孩子，自然就不考虑它们这一人群的性别了）。而且丹丹了解到：如果三个女孩排在一起，她们就会互相攀比；如果三个男孩排在一起，他们就会喋喋不休；如果三个熊孩子排在一起，哦我的上帝！天都要塌下来了！一旦出现这种情况，丹丹就会为这些吵闹的人感到惋惜而无心过生日。而她的好伙伴们则希望丹丹总是开开心心，所以他们接着去分析这个问题。

他们注意到，除了上述所讲的三种情况之外，还有不少男生女生是借丹丹的生日派对之便为趁机搭讪而来的。所以如果派对的过程中，男生和女生之间夹有一个熊孩子，这样这对少男少女就会十分无奈，丹丹也会为此惋惜而变得郁郁寡欢。另外，如果男生或者女生被两个熊孩子加在中间，熊孩子迟早会把那个倒霉的人惹烦。这样一来，丹丹会觉得她办了一个失败的生日派对而郁郁寡欢。

现在丹丹的好伙伴们跑来询问你，针对一个排了 N 人的队伍，到底有多少种让丹丹满意的排列方案？由于答案可能会很庞大，所以他们麻烦你将答案对 1,000,000,007 取模后，再告诉他们。

输入数据

有十组测试数据。每一组的输入数据格式相同。

第一行是一个整数 T ，表示有 T 组测试样例。

之后的 T 行中，每一行有一个数字 N ，表示这个队伍的总人数。

输出数据

针对每一个测试样例，输出一行结果。

结果的格式为“Case #No: Ans”，No 表示第 No 组样例，Ans 表示该组样例的答案。

具体的请参见输出样例。

测试样例

样例输入	样例输出
3	Case #1: 20
3	Case #2: 46
4	Case #3: 435170
15	

数据规模

对于所有的数据， T 满足 $1 \leq T \leq 1500$ 。

对于 30% 的数据，每组 3 分，保证 N 满足 $0 \leq N \leq 10^4$ 。

对于 30% 的数据，每组 7 分，保证 N 满足 $0 \leq N \leq 10^7$ 。

对于 20% 的数据，每组 15 分，保证 N 满足 $0 \leq N \leq 10^9$ 。

对于 20% 的数据，每组 20 分，保证 N 满足 $0 \leq N \leq 10^{11}$ 。

B. 丹丹的数学问题

时间限制	1s
空间限制	64MB
源程序名	math.{c,cpp,java}
输入文件	math.in
输出文件	math.out

问题描述

数学作业总是最烦人的一项，不论何时，不论何地。丹丹的数学作业还没写完，但是她又想在生日派对上好好地轻松一天，所以她把她的数学作业就交给你了。

其实作业只有一道题目，讲的是一个正整数数列 $1, 2, \dots, n$ ，每一次都可以选取任意多个数列中的数据进行一次操作，让这些被选中的数字统一减去一个相同的正整数，但是不能减成负数。目标是将整个数列的数通过上述操作全部变成 0。求最少的操作次数。

“这个问题太简单了！”你这样想着，“告诉我这个数列 n 的大小，我来告诉你答案。”

输入数据

有十组测试数据。每一组的输入数据格式相同。

第一行是一个整数 T ，表示有 T 组测试样例。

之后的 T 行中，每一行有一个数字 N ，表示这个数列的长度。

输出数据

针对每一个测试样例，输出一行结果。

结果的格式为“Case #No: Ans”，No 表示第 No 组样例，Ans 表示该组样例的答案。

具体的请参见输出样例。

测试样例

样例输入	样例输出
3	Case #1: 1
1	Case #2: 2
2	Case #3: 2
3	

数据规模

对于 20% 的数据，每组 2 分，保证 T 满足 $1 \leq T \leq 10^2$ ，保证 N 满足 $1 \leq N \leq 10^3$ 。

对于 30% 的数据，每组 4 分，保证 T 满足 $1 \leq T \leq 10^3$ ，保证 N 满足 $1 \leq N \leq 10^6$ 。

对于 20% 的数据，每组 15 分，保证 T 满足 $1 \leq T \leq 10^4$ ，保证 N 满足 $1 \leq N \leq 10^8$ 。

对于 30% 的数据，每组 18 分，保证 T 满足 $1 \leq T \leq 10^6$ ，保证 N 满足 $1 \leq N \leq 10^9$ 。

C. 丹丹的生日礼物

时间限制	3s
空间限制	256MB
源程序名	prize.{c,cpp,java}
输入文件	prize.in
输出文件	prize.out

问题描述

丹丹的好伙伴们为丹丹提前准备好了生日礼物，当丹丹看到这么多的礼物的时候，不禁感叹：“太多了！你们太好了！”

丹丹不想把所有的礼物全部拿走，因为在这些礼物中，有的太贵重了，她不想独占所有的礼物，她想这份开心分享给他的好伙伴们。所以她决定只拿走总价值为 K 的礼物。

丹丹的好伙伴们总共为丹丹准备了 N 种不同的礼物，每种礼物有 4 中不同的样式。对每一种不同的礼物，丹丹都可以选取任意件数的该种礼物，只要拿走的数量不超过这种礼物的总数 4 件就行。

现在丹丹想知道，她到底能不能拿走总价值恰好为 K 的礼物。

输入数据

有十组测试数据。每一组的输入数据格式相同。

第一行是一个整数 T ，表示有 T 组测试样例。

每组数据的第一行是两个整数 N 和 K ，第二行会有 N 个数，分别是送来的礼物的价值 w_i 。

输出数据

针对每一个测试样例，输出一行结果。

结果的格式为“Case #No: Ans”，No 表示第 No 组样例，Ans 表示该组样例的答案。如果可以取走礼物输出 yes，否则输出 no。

具体的请参见输出样例。

测试样例

样例输入	样例输出
3	Case #1: yes
2 5	Case #2: no
1 2	Case #3: yes
2 10	
1 2	
3 10	
1 3 5	

数据规模

对于所有的数据， T 满足 $1 \leq T \leq 10$ 。

对于 10% 的数据，每组 2 分，保证 N 满足 $1 \leq N \leq 5$ ， K 满足 $1 \leq K \leq 10^9$ ， w_i 满足 $1 \leq w_i \leq 10^3$ 。

对于 20% 的数据，每组 4 分，保证 N 满足 $1 \leq N \leq 9$ ， K 满足 $1 \leq K \leq 10^9$ ， w_i 满足 $1 \leq w_i \leq 10^4$ 。

对于 30% 的数据，每组 10 分，保证 N 满足 $1 \leq N \leq 15$ ， K 满足 $1 \leq K \leq 10^9$ ， w_i 满足 $1 \leq w_i \leq 10^5$ 。

对于 40% 的数据，每组 15 分，保证 N 满足 $1 \leq N \leq 18$ ， K 满足 $1 \leq K \leq 10^9$ ， w_i 满足 $1 \leq w_i \leq 10^6$ 。

D. 丹丹的路边向导

时间限制	3s
空间限制	64MB
源程序名	guide.{c,cpp,java}
输入文件	guide.in
输出文件	guide.out

问题描述

为了方便来参加丹丹的生日派对的大家准确的找到丹丹的家，丹丹和她的好伙伴们想到了一个方法，那就是在路口放置上路标，这样大家就能找到路了。但是在每一个路口都放置上路标显然是不可能的，丹丹不希望她的好伙伴们为此太费心费力，所以丹丹的好伙伴们把这个问题再次简化。由于大家都是来丹丹的家里参加她的生日派对，所以相当于从丹丹的家发散出了一个道路网。而且，如果一个路口上有一个路标，与这个路口相连的道路都能看到路标上的信息。为了让所有的人能够正确的找到丹丹的家，丹丹希望能够用最少的路标，来保证大家都能找到丹丹的家。

输入数据

有十组测试数据。每一组的输入数据格式相同。

第一行是一个整数 T ，表示有 T 组测试样例。

每组数据的第一行是一个整数 N ，表示有 N 个路口。随后的 N 行中，每一行都有至少两个整数。第一个整数 i 表示这个路口的标号（从 0 开始标记，到 $N-1$ 为止）。第二个整数 d 表示这个路口所连接的新的路口数量。后面的 d 个整数分别表示这个路口所连接的新的路口的标号。如果第二个整数 $d=0$ ，你可以认为这是路口恰好是一个要来参加派对的人的住址。

输出数据

针对每一个测试样例，输出一行结果。

结果的格式为“Case #No: Ans”，No 表示第 No 组样例，Ans 表示该组样例的答案，即最少的路标放置数量。

具体的请参见输出样例。

测试样例

样例输入	样例输出
2	Case #1: 1
4	Case #2: 2
0 1 1	
1 2 2 3	
2 0	
3 0	
5	
3 3 1 4 2	
1 1 0	
2 0	
0 0	
4 0	

样例解释

对于第 1 组样例，在 1 号路口放置路标即可。

对于第 2 组样例，可以在 1 号路口和 3 号路口放置路标，或者在 0 号和 3 号放置路标。以上两种方案均可。

数据规模

对于所有的数据， T 满足 $50 \leq T \leq 100$ 。

对于 10% 的数据，每组 2 分，保证 N 满足 $1 \leq N \leq 100$ 。

对于 20% 的数据，每组 4 分，保证 N 满足 $1 \leq N \leq 500$ 。

对于 30% 的数据，每组 10 分，保证 N 满足 $1 \leq N \leq 4000$ 。

对于 40% 的数据，每组 15 分，保证 N 满足 $1 \leq N \leq 8000$ 。

E. 丹丹的合影留念

时间限制	1.5s
空间限制	64MB
源程序名	photo.{c,cpp,java}
输入文件	photo.in
输出文件	photo.out

问题描述

丹丹的生日派对就要结束了，“来的人真多”，丹丹这样想着，“大家一起合个影留个念吧！”

但是事情总不是那么轻轻松松简简单单，丹丹希望每个人在合影时都有一个位置，不至于到最后照片上只有半张脸。在丹丹和她的好伙伴们讨论之后，他们决定先把人一个一个的站好，排列成一个 $M \times N$ 矩形，即有 M 行，每行 N 个人的矩形方阵。排好队之后再拍照留念。

这个时候问题来了（不是挖掘机我谢谢你的自动脑补 =。=），人数不会总是那么巧合正好可以按照这个方法排列成一个矩形方阵。所以他们想到了一些简单的挽救方案。允许中间留空，但是为了整体的美观，不允许在整个矩形的边界出现空白，也不允许一个空白的周围（前、后、左、右、左前、右前、左后、右后）出现空白。那么如果现在告诉你当前准备排列的行数 M 与每行的人数 N ，以及总的人数 K ，能不能告诉丹丹和她的好伙伴们能不能排列出一个合适的矩形，从而让她们成功合影。

输入数据

有十组测试数据。每一组的输入数据格式相同。

第一行是一个整数 T ，表示有 T 组测试样例。

之后的 T 行中，每一行有三个数字 M, N, K ，分别表示矩形方阵的行数，每行的人数，和总人数。

输出数据

针对每一个测试样例，输出一行结果。

结果的格式为“Case #No: Ans”，No 表示第 No 组样例，Ans 表示该组样例的答案。如果可以排列成 $M \times N$ 的矩形方阵，输出 yes，否则输出 no。

具体的请参见输出样例。

测试样例

样例输入	样例输出
3	Case #1: yes
3 3 9	Case #2: yes
3 3 8	Case #3: no
3 3 3	

数据规模

对于所有的数据， T 满足 $150 \leq T \leq 200$ 。

对于 10% 的数据，每组 2 分，保证 N, M 满足 $1 \leq N, M \leq 10^6$ ， K 满足 $1 \leq K \leq 10^{13}$

对于 20% 的数据，每组 4 分，保证 N, M 满足 $1 \leq N, M \leq 10^{12}$ ， K 满足 $1 \leq K \leq 10^{25}$

对于 30% 的数据，每组 10 分，保证 N, M 满足 $1 \leq N, M \leq 10^{1000}$ ， K 满足 $1 \leq K \leq 10^{1000}$

对于 40% 的数据，每组 15 分，保证 N, M 满足 $1 \leq N, M \leq 10^{1500}$ ， K 满足 $1 \leq K \leq 10^{1500}$