

# GDSY-PION 模拟赛

	dp	线段树	博弈论	图论
输入文件名	dp.in	segtree.in	game.in	graph.in
输出文件名	dp.out	segtree.out	game.out	graph.out
源程序文件名	dp.cpp	segtree.cpp	game.cpp	graph.cpp
时间限制	1s	3s	2s	2s
空间限制	64MB	512MB	512MB	256MB
题目类型	传统	传统	传统	传统
结果比较方式	全文比较	全文比较	全文比较	全文比较
是否捆绑测试	否	否	否	是

编译选项：

| 对于 C++ 语言 | -O2 -std=c++14 -static |  
| :: | :: |

## 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. 提交的程序代码文件的放置位置请参照具体需求。
4. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
5. 选手提交的程序源文件必须不大于 100KB。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。

# dp

输入: dp.in

输出: dp.out

时间限制: 1s

空间限制: 64MB

## 题目背景

由于数据问题，请参赛选手在读入  $n = 0$  时什么都不要输出，跳过这一组询问。

## 题目描述

有一个单人跳棋游戏，规则如下：

假设有一个无限大的棋盘，棋盘上有一些棋子。一个棋子可以跳过并吃掉和它相邻的一枚棋子，这是棋子的唯一一种移动方式。

形式化定义如下：每次操作可以选择一个棋子  $(x, y)$ ，并从以下四种行动中选择一种：

1. 若位置  $(x - 1, y)$  存在棋子，则可以将当前棋子移动至  $(x - 2, y)$ ，并移除处于  $(x - 1, y)$  的棋子。
2. 若位置  $(x, y - 1)$  存在棋子，则可以将当前棋子移动至  $(x, y - 2)$ ，并移除处于  $(x, y - 1)$  的棋子。
3. 若位置  $(x + 1, y)$  存在棋子，则可以将当前棋子移动至  $(x + 2, y)$ ，并移除处于  $(x + 1, y)$  的棋子。
4. 若位置  $(x, y + 1)$  存在棋子，则可以将当前棋子移动至  $(x, y + 2)$ ，并移除处于  $(x, y + 1)$  的棋子。

现在，在某个位置画一条无限长的水平线，你需要在水平线下面放置足够多的棋子，使得它们“前仆后继”地往水平线上方跳，最终能够跳到水平线以上  $n$  个单位的位置。

现在输入  $n$ ，询问至少需要多少个棋子。如果无解，输出  $-1$ 。

## 输入格式

有多组测试数据，输入格式如下：

第一行一个整数  $T$ ，代表数据组数。

接下来  $T$  行每行一个整数  $n$ ，代表本组询问的距离。

## 输出格式

输出  $T$  行，每行一个整数，表示这组询问的答案

## 样例

样例输入

```
1 | 3
2 | 1
3 | 4
4 | 9999
```

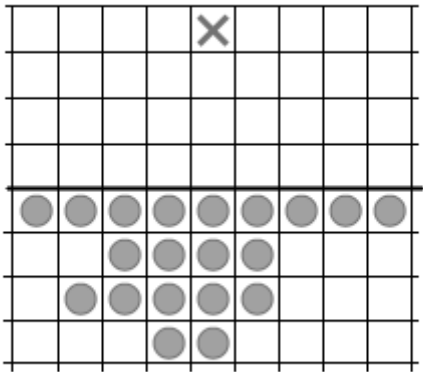
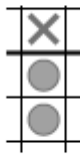
样例输出

1	2
2	20
3	-1

说明提示

样例说明

样例的棋子初始排列方式（之一）如图



数据规模与约定

- 对于 30% 的数据，满足  $n \leq 100, t \leq 100$  .
- 对于 100% 的数据，满足  $n \leq 10000, t \leq 10000$  .

# 线段树

输入: segtree.in

输出: segtree.out

时间限制: 3s

空间限制: 512MB

## 题目背景

线段树??

## 题目描述

你现在有一个长度为  $n$  ( $n > 2$ ) 的数组  $A$ , 其下标范围为  $[0, n-1]$ , 初始每一位上都是 0.

数组上有两个指针  $x$  和  $y$ ,  $x$  初始为 0,  $y$  初始为  $n-2$ .

有八种操作, 按以下格式给出。

- 1 k, 将  $x$  的值改为  $x+1$ , 若修改后  $x=y$  就将  $x$  修改为 0. (此时保证  $k=0$ )
- 2 k, 将  $y$  的值改为  $y-1$ , 若修改后  $x=y$  就将  $y$  修改为  $n-2$ . (此时保证  $k=0$ )
- 3 k, 对于所有的  $i \in [0, x]$ , 将  $A_i$  加上  $k$  ( $k$  可能为负数).
- 4 k, 对于所有的  $i \in [0, y]$ , 将  $A_i$  加上  $k$  ( $k$  可能为负数).
- 5 k, 对于所有的  $i \in [0, x]$ , 求  $A_i$  的最小值. (此时保证  $k=0$ )
- 6 k, 对于所有的  $i \in [0, y]$ , 求  $A_i$  的最小值. (此时保证  $k=0$ )
- 7 k, 对于所有的  $i \in [x+1, n-1]$ , 求  $A_i$  的最小值. (此时保证  $k=0$ )
- 8 k, 对于所有的  $i \in [y+1, n-1]$ , 求  $A_i$  的最小值. (此时保证  $k=0$ )

## 输入格式

为减少输入量, 输入格式如下:

第一行包含两个正整数  $n, m$ , 分别表示数列  $A$  的长度和操作次数.

第二行给出 5 个正整数  $Seed, W, X, Y, Z$ .

你应当将如下代码加入你代码的开头:

```
1 unsigned long long Seed;
2 long long w,x,y,z;
3 unsigned long long HRAND()
4 {
5     Seed^=Seed<<9;
6     Seed^=Seed>>11;
7     Seed^=Seed<<15;
8     return Seed;
9 }
10 void Gen(int &op,int &k)
11 {
12     long long OP=HRAND()%(X+Y+2*Z);
13     if(OP<X)          op=1+HRAND()%2,k=0;
14     else if(OP<X+Y)    op=3+HRAND()%2,k=HRAND()%(2*w)-w;
15     else if(OP<X+Y+Z)  op=5+HRAND()%2,k=0;
16     else               op=7+HRAND()%2,k=0;
```

在每次操作时，你应当调用一次  $Gen$  函数，以获得本次的操作编号及操作参数  $op, k$ 。

## 输出格式

为减少输出量，输出格式如下：

初始化  $ans$  为 0。

在每次询问操作时，设  $B_i$  为第  $i$  次操作的结果，则将  $ans$  修改为  $ans \otimes ((i\%998) * (B_i + 5e14))$ 。

所有操作结束后输出  $ans$ 。

## 样例

### 样例输入

```
1 4 10
2 1 52 2 2 1
```

### 样例输出

```
1 2504982447374640
```

其余样例见附加文件

## 说明提示

### 样例说明

第 $i$ 次操作后	$x$	$y$	$A$
1	0	2	$-47, -47, -47, 0$
2	0	2	$-29, -29, -29, 0$
3	0	1	$-29, -29, -29, 0$
5	0	1	$-29, -29, -29, 0$
6	0	2	$-29, -29, -29, 0$
7	0	1	$-29, -29, -29, 0$
8	0	1	$-36, -36, -29, 0$
10	0	2	$-36, -36, -29, 0$

两次的结果分别为： $-29, -36$ 。

所以最终答案为  $(2e15 - 4 * 29) \otimes (45e14 - 9 * 36) = 2504982447374640$

## 数据规模与约定

- 对于 30% 的数据，满足  $n \leq 1e6, m \leq 2e6$  .
- 对于 100% 的数据，满足  $n \leq 1e7, m \leq 5e7, |k| \leq 1e7$  .

## 提示

别线段树啦，过不去的.

~~代码里尽量不要直接写  $5e14$ ，因为  $5e14$  在代码里可能会识别为浮点数，导致出错.~~

# 博弈论

输入: game.in

输出: game.out

时间限制: 2s

空间限制: 512MB

## 题目背景

博弈论 ???

## 题目描述

Alice 有  $n$  个外形完全相同的球，第  $i$  个球里写了数字  $a_i$ 。

Alice 和 Bob 玩游戏。

Alice 选出两个球  $i, j$ ，满足不存在  $k$  使得  $(a_i - a_k)(a_j - a_k) < 0$ （即不存在数字严格夹在中间的球），然后把它们分别放在 Bob 的左右手。

Bob 的眼睛被蒙住了，他并不知道 Alice 选了哪两个球。

Bob 需要选择“左手”或“右手”。如果选择了左手，那么 Alice 会告诉 Bob  $a_i$ ，然后 Bob 需要猜测  $a_i, a_j$  的大小关系是  $>, =, <$  的哪一种。

同理如果选择了右手则 Alice 会告诉 Bob  $a_j$ ，Bob 仍需猜测大小关系。若 Bob 猜对了则 Bob 获胜，否则 Alice 获胜。

帮 Bob 选择一个最优的随机策略，求出 Bob 能获得的最大胜率  $p$ ，使得无论 Alice 选的是哪两个球，胜率都至少是  $p$ 。答案模 998244353 输出。

## 输入格式

有多组测试数据，输入格式如下：

第一行一个整数  $T$ ，代表数据组数。

接下来  $T$  行每行六个整数  $n, a_0, A, B, C, M$ 。 $a_1, \dots, a_n$  的生成方式如下

$$a_i = ((A \cdot a_{i-1}^2 + B \cdot a_{i-1} + C) \bmod M) + 1$$

## 输出格式

输出  $T$  行，每行一个整数，表示这组询问模 998244353 的答案。

## 样例

样例输入

```
1 2
2 4 1 0 1 0 2
3 4 1 0 1 0 4
```

样例输出

1	499122177
2	665496236

说明提示

样例说明

对于第一组数据， $n = 4$ ， $a_1, a_2, a_3, a_4 = 2, 1, 2, 1$ 。  
Bob 的一个最优策略是：报出左手，然后在 1 和 2 间等概率随机猜测。  
对于第二组数据，答案是  $\frac{2}{3}$ 。

数据规模与约定

- 对于 15% 的数据，满足  $\sum n \leq 30$ 。
- 对于 30% 的数据，满足  $\sum n \leq 300$ 。
- 对于 50% 的数据，满足  $\sum n \leq 5000$ 。
- 对于 70% 的数据，满足  $\sum n \leq 10^6$ 。
- 对于 100% 的数据，满足  $T \leq 1000, \sum n \leq 10^7, 0 \leq a_0, A, B, C \leq 10^9, 1 \leq M \leq n$ 。



# 图论

输入: graph.in

输出: graph.out

时间限制: 2s

空间限制: 256MB

## 题目背景

图论 ? ? ? ?

## 题目描述

很久以前,有一个复杂的遗迹,可以抽象为一个无向图。已知这个图有  $n$  个点,编号为  $1 \dots n$ ,但是你并不知道共有几条边。记每第  $i$  个点的度数为  $d_i$ 。

为了探索这个图,你发明了一种自分裂机器人。初始时,你将这个机器人放在点 1。

每一时刻,每个机器人会恰好分裂为  $d_i$  个一样的机器人,这  $d_i$  个机器人会分别前往  $d_i$  与当前点相邻的节点。

由于信号干扰,你并无法得知机器人的具体分裂情况。但是你可以在每一时刻探测每个节点上是否有机器人。

根据收集到的信息,你的电脑自动的给出了一种可能的遗迹图像。这个图有  $m$  条边。**注意:遗迹的实际边数不一定是  $m$ 。**

看着这个可能的遗迹图,你突然想知道总共有多少种可能的遗迹图像。

## 输入格式

本题有多组测试数据,每组测试数据间独立。

输入的第一行包含  $T$ ,为测试用例的数量。

每个测试用例的第一行包含整数  $n$  和  $m$ ,分别表示遗迹的点数,以及一个可能的遗迹图的边数。

每个测试用例的以下  $m$  行每行包含两个整数  $x$  和  $y$  ( $1 \leq x \leq y \leq N$ ),表示电脑给出的图中存在一条连接  $x$  与  $y$  的边。

为提高可读性,相邻的测试用例之间用一个空行隔开。

## 输出格式

对每个测试用例,输出一行,为不同的可能的遗迹图的数量,对  $10^9 + 7$  取模。

## 样例

样例输入 #1

1	1
2	
3	5 4
4	1 2
5	2 3
6	1 4
7	3 5

### 样例输出 #1

1	3
---	---

其余样例见附加文件

## 说明提示

### 样例说明

在第一个测试用例中，遗迹图，可能为以下三个图之一：

1	5 4
2	1 2
3	2 3
4	1 4
5	3 5
6	
7	5 4
8	1 2
9	1 4
10	3 4
11	3 5
12	
13	5 5
14	1 2
15	2 3
16	1 4
17	3 4
18	3 5

### 样例输出 #2

1	7
2	
3	4 6
4	1 2
5	2 3
6	3 4
7	1 3
8	2 4
9	1 4
10	
11	5 5
12	1 2
13	2 3
14	3 4

15	4 5
16	1 5
17	
18	5 7
19	1 2
20	1 3
21	1 5
22	2 4
23	3 3
24	3 4
25	4 5
26	
27	6 6
28	1 2
29	2 3
30	3 4
31	4 5
32	5 6
33	6 6
34	
35	6 7
36	1 2
37	2 3
38	1 3
39	1 4
40	4 5
41	5 6
42	1 6
43	
44	10 10
45	1 1
46	1 2
47	1 3
48	1 4
49	1 5
50	1 6
51	1 7
52	1 8
53	1 9
54	1 10
55	
56	22 28
57	1 2
58	2 3
59	3 4
60	4 5
61	5 6
62	6 7
63	1 7
64	1 8
65	3 9
66	8 10
67	10 11
68	10 12
69	10 13
70	10 14
71	11 15
72	12 16

73	13	17
74	14	18
75	9	15
76	9	16
77	9	17
78	9	18
79	15	19
80	19	20
81	15	20
82	16	21
83	21	22
84	16	22

## 样例输出 #2

1	45
2	35
3	11
4	1
5	15
6	371842544
7	256838540

### 数据规模与约定

- 对于 10% 的数据，满足  $n \leq 5$  .
- 对于另外 10% 的数据，满足  $m = n - 1$  .
- 对于 100% 的数据，满足  $n \leq 100$  .

数据较水，欢迎暴力。