

2018 山东省队集训

第一轮

Day 1

时间：2018 年 6 月 9 日 08:00 ~ 13:00

题目名称	开车	上分	隔膜
题目类型	传统型	传统型	传统型
目录	carcar	lose	game
可执行文件名	carcar	lose	game
输入文件名	carcar.in	lose.in	game.in
输出文件名	carcar.out	lose.out	game.out
每个测试点时限	1.0 秒	4.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
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	carcar.cpp	lose.cpp	game.cpp
对于 C 语言	carcar.c	lose.c	game.c
对于 Pascal 语言	carcar.pas	lose.pas	game.pas

编译选项

对于 C++ 语言	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11
对于 C 语言	-O2	-O2	-O2
对于 Pascal 语言	-O2	-O2	-O2

开车 (carcar)

【题目描述】

作为钦钦草原最绿的男人，杨某针每天都要开车巡视钦钦草原一圈。

钦钦草原由 n 个城市组成， m 条双向道路连接着它们。经过第 i 条道路要花费的时间是 2^i 。

杨某针想要经过每条道路至少一次，在此基础上他想最小化他花费的时间。但作为曾经 CTSC 的 Cu 选手，他并不能很快地计算出这个问题。所以他向你求助。

【输入格式】

从文件 *carcar.in* 中读入数据。

输入第一行包含两个正整数 n, m 。

接下来 m 行，每行两个正整数 a_i, b_i ，表示第 i 条边连接点 a_i 和 b_i ，它的权值为 2^i 。

保证 $a_i \neq b_i$ ，不存在重边，且任意两个点之间可以互相到达。

【输出格式】

输出到文件 *carcar.out* 中。

输出一行一个整数，表示答案对 $10^9 + 7$ 取模的值。

【样例 1 输入】

4 5

1 2

3 4

2 3

1 3

2 4

【样例 1 输出】

70

【样例 1 解释】

最优的路线应当为 1-2-3-4-2-3-1。

【样例 2 输入】

```
6 10
4 6
4 5
3 6
5 2
3 2
1 2
3 4
6 1
2 4
1 3
```

【样例 2 输出】

```
2132
```

【子任务】

本题使用捆绑测试。每个子任务有若干个测试点，分为 3 个子任务，你只有通过一个子任务的所有测试点才能得到这个子任务的分数。

对于 20% 的数据， $n, m \leq 20$ 。

对于 40% 的数据， $n, m \leq 2,000$ 。

对于 100% 的数据， $n \leq 400,000, m \leq 500,000$ 。

上分 (lose)

【题目描述】

作为钦钦草原最绿的男人，杨某针决定在某 dead game 里提升一下自己的水平。

这款 dead game 一共有 10^6 关，对于其中的第 i 关，它又有恰好 i 个小关。我们记第 i 关的第 j 个关卡是 (i, j) 。

小针针如果想要通过 (i, j) 这一关卡，那么他首先要通过 $(i-1, j-1)$ 和 $(i-1, j)$ 这两个关卡。（如果这两个关卡存在的话）

现在，小针针想要知道，如果他想要通过 (A, B) 和 (C, D) 这两个关卡，在通过的关卡总数最小的情况下，他有多少种本质不同的通关方法？两种方法被认为是本质不同的当且仅当某一次这两种方法通过的关卡不同。

杨某针一共会被询问 T 次。作为曾经 APIO 的国际 Au 选手，他当然能轻松地解决这个问题。但是他打算考考你。

因为杨某针的精力有限，所以保证通过 (A, B) 和 (C, D) 这两个关卡的所需要的最少通过关卡总数不超过 10^6 。

【输入格式】

从文件 *lose.in* 中读入数据。

输入第一行包含一个正整数 T 。

接下来 T 行，每行四个正整数 A_i, B_i, C_i, D_i ，表示一组询问。

保证 $1 \leq B_i \leq A_i \leq 10^6, 1 \leq D_i \leq C_i \leq 10^6$ 。

保证 (A_i, B_i) 与 (C_i, D_i) 不同。

【输出格式】

输出到文件 *lose.out* 中。

输出 T 行，每行一个整数，表示答案对 $10^9 + 7$ 取模的值。

【样例 1 输入】

```
6
2 1 2 2
1 1 1000000 1000000
3 2 5 3
5 2 4 3
2015 55 1700 1300
100 50 1000 500
```

【样例 1 输出】

2
1
42
252
926737422
143485143

【样例 1 解释】

对于第一组询问，有两种方案。它们分别为：

$(1, 1), (2, 2), (2, 1)$

$(1, 1), (2, 1), (2, 2)$

【子任务】

本题使用捆绑测试。每个子任务有若干个测试点，分为 3 个子任务，你只有通过一个子任务的所有测试点才能得到这个子任务的分数。

对于 20% 的数据， $A_i, B_i, C_i, D_i \leq 4$ 。

对于另外 20% 的数据， $T = 1$ 。

对于 100% 的数据， $T \leq 300,000$ 。

隔膜 (game)

【题目描述】

阿针和阿吨在玩一个奇怪的游戏。

n 个硬币排成一排，第 i 个硬币的价值是 S_i 。最开始，奇数位置上的硬币是正面朝上的，偶数位置上的硬币是反面朝上的。

游戏一共有 $n - 1$ 轮，把它们从 1 到 $n - 1$ 标号。阿针可以在奇数轮行动，而阿吨可以在偶数轮行动。在第 i 轮，他们可以选择翻转 i 或者 $i + 1$ 中的某一枚硬币或者什么也不做。

当游戏结束后，阿针的得分是正面朝上的硬币价值之和，而阿吨的得分是反面朝上的硬币价值之和。他们都想让自己的得分最大化。

现在，阿针提出了 Q 个修改，其中第 i 个修改是把硬币 P_i 的价值变小了 D_i 。在每次修改后，他都想知道他能拿到多少分。

作为绝活狼人上分哥，阿针当然会这个问题了，但他决定考考你，不然你就会被他速推了。

【输入格式】

从文件 `game.in` 中读入数据。

输入第一行包含一个正整数 n 。

第二行 n 个正整数 S_i 。

接下来一行一个正整数 Q 。

接下来 Q 行，每行两个正整数 P_i, D_i ，表示一次修改。

保证 $2 \leq N$ 。

保证 $1 \leq S_i \leq 10^9$ 。

保证 $1 \leq P_i \leq N, 1 \leq D_i$ 。

保证任何时刻每个硬币的价值都是正数。

【输出格式】

输出到文件 `game.out` 中。

输出 $Q + 1$ 行。

第 i 行输出在第 $i - 1$ 次修改后阿针的期望得分。

【样例 1 输入】

4

1 2 3 4

1
4 3

【样例 1 输出】

7
5

【样例 2 输入】

8
3 1 4 1 5 9 2 6
5
3 3
6 6
8 4
1 1
6 2

【样例 2 输出】

19
16
16
12
11
11

【子任务】

本题使用捆绑测试。每个子任务有若干个测试点，分为 3 个子任务，你只有通过一个子任务的所有测试点才能得到这个子任务的分数。

对于 20% 的数据， $n, q \leq 20$ 。

对于 40% 的数据， $n, q \leq 2,000$ 。

对于 100% 的数据， $n, q \leq 200,000$ 。