

题目名称	魔法师 (wizard.cpp)	二叉搜索树 (bst.cpp)	枚举 (enum.cpp)	树 (tree.cpp)
输入文件名	wizard.in	bst.in	enum.in	tree.in
输出文件名	wizard.out	bst.out	enum.out	tree.out
时限	1s	1s	2s	6s
内存	512M	512M	512M	1G
编译命令	-O2	-O2	-O2	-O2

# 魔法师(wizard)

## 问题描述

小M正在玩一个冒险游戏。游戏中，他是一名魔法师，需要在一片神秘的森林中使用法术击败各种怪物。

击败每个怪物都会给予小M一种法术，但是小M只能同时保持一种法术，如果击败新的怪物会遗忘之前的法术。当然，使用不同的法术击败不同的怪物需要不同的回合数。每次使用法术都需要消耗一定的法力值，小M的总法力值是有限的。

他拥有一个道具，这使得他击败第一个怪物不需要时间和法力值。

现在小M希望以最快的速度击败所有怪物，但是他不知道怎么样操作最快。请你帮帮他。

## 输入格式

输入共  $2n + 1$  行。第一行两个整数  $n$  和  $m$ ，分别表示怪物的数量和小M的法力值。

接下来  $n$  行，其中每行  $n$  个整数，第  $i$  行第  $j$  个整数记为  $w_{i,j}$ ，表示使用怪物  $i$  提供的法术击败怪物  $j$  需要的回合数。本题中出现的怪物编号都是从 1 到  $n$  的正整数。

接下来  $n$  行，其中每行  $n$  个整数，第  $i$  行第  $j$  个整数记为  $c_{i,j}$ ，表示使用怪物  $i$  提供的法术击败怪物  $j$  需要的法力值。

## 输出格式

如果无解，输出一行一个整数  $-1$ 。

在有解的情况下，输出共 2 行。第 1 行一个整数，表示在小M的法力值允许的情况下击败所有怪物的最短回合数。第 2 行为  $n$  个整数的编号序列，第  $i$  个整数表示小M击败的第  $i$  个怪物编号（含起点与终点）。如果存在多个最优解，请输出使得该编号序列字典序最小的一条。

注意，“字典序比较”是指“逐点比较”而非“逐字符比较”，如 2, 10 的字典序小于 10, 2。

## 样例输入

```
4 18
0 9 8 5
7 0 2 3
6 4 0 8
1 7 1 0
0 8 8 8
4 0 8 6
8 5 0 8
9 3 5 0
```

## 样例输出

```
10
1 4 3 2
```

## 样例说明

操作顺序为  $1 \rightarrow 4 \rightarrow 3 \rightarrow 2$ ，总回合数为  $5 + 1 + 4 = 10$ ，法力总花费为  $8 + 5 + 5 = 18$ ，是法力不超过上限的最快过程。

值得注意的是，虽然存在更快速的操作顺序  $3 \rightarrow 2 \rightarrow 4 \rightarrow 1$ ，总回合数为  $4 + 3 + 1 = 8$ ，但是其法力花费为  $5 + 6 + 9 = 20$ ，超过了法力上限。

## 数据范围与约定

对于 10% 的数据， $n \leq 5$ 。

对于 30% 的数据， $n \leq 10$ 。

对于 50% 的数据， $n \leq 13$ 。

对于 70% 的数据， $n \leq 16$ 。

对于 100% 的数据， $3 \leq n \leq 18, 0 \leq m \leq 10^9$ 。对任意  $i \leq j$ ，有  $0 \leq w_{i,j} \leq 10^6, 0 \leq c_{i,j} \leq 10^6$ ， $w_{i,i}$  和  $c_{i,i}$  的值在本题中无意义，输入数据中取值为 0。所有的  $w_{i,j}, c_{i,j}$  在数据范围内随机生成， $m$  以一条随机操作序列的总花费为基础，在其附近随机取值。

# 二叉搜索树(bst)

### 题目描述

给出一个  $n$  个点，以 1 为根的二叉树，其中每个点有个权值。有  $q$  次修改，每次修改一个点的点权，你需要在每次修改之后输出有多少个点  $u$ ，满足以  $u$  为根的子树是一棵二叉搜索树。

一棵树是二叉搜索树当且仅当对所有点  $x$  均满足以下两个条件：

- 不存在左子树或左子树中的所有点的点权都  $\leq x$  的点权
- 不存在右子树或右子树中的所有点的点权都  $\geq x$  的点权

输入格式

第一行两个正整数  $n, q$  。

接下来  $n$  行，第  $i + 1$  行两个数  $l_i, r_i$ ，代表点  $i$  的左儿子和右儿子的标号，如果不存在则为 0。

接下来一行  $n$  个数  $v_1, v_2, \cdots, v_n$ ，代表  $n$  个点初始的点权。

接下来  $q$  行每行两个数  $k_i, x_i$ ，代表这次修改将点  $k_i$  的权值改为  $x_i$ 。

输出格式

一共  $q$  行，每行包含一个正整数，第  $i$  行的正整数代表第  $i$  次修改之后的答案。

样例输入 1

```
6 5
2 3
4 0
5 6
0 0
0 0
0 0
4 1 3 2 2 5
3 3
2 2
3 5
5 4
6 1
```

样例输出 1

```
4
5
5
6
4
```

样例输入 2

```
8 10
4 5
8 0
0 0
3 7
0 6
0 0
2 0
0 0
7 0 9 3 6 0 6 2
3 0
4 0
8 2
2 3
7 6
```

```
1 6
5 7
6 9
1 1
1 7
```

## 样例输出 2

```
3
3
3
6
6
6
6
8
7
8
```

## 数据范围

对于10%的数据,  $n, q \leq 500$ 。

对于 20% 的数据,  $n, q \leq 5000$ 。

对于另外 30% 的数据, 保证固定点数后, 树的构造方式为: 以 1 为根, 在  $[0, \text{当前子树需要的大小} - 1]$  中均匀随机, 并其设为左子树的大小, 递归到左子树和右子树继续构造。

对于100%的数据,  $1 \leq n, q \leq 2 * 10^5, 0 \leq l_i, r_i \leq n, 1 \leq k_i \leq n, 0 \leq v_i, x_i \leq 10^9$ 。

# 枚举 (enum)

## 问题描述

小A给出  $2^k - 1$  个数  $a_1, a_2, \dots, a_{2^k-1}$ 。

小B给出  $k$  个数  $b_0, b_1, \dots, b_{k-1}$ 。

而你是小C, 需要给出  $2^k - 1$  个数  $c_1, c_2, \dots, c_{2^k-1}$ , 满足对于所有  $j = 0 \dots k - 1$ :

$$\sum_{i=1}^{2^k-1} (\lfloor \frac{i}{2^j} \rfloor \bmod 2) * c_i = b_j$$

成立, 你要求出所有情况下  $\sum_{i=1}^{2^k-1} a_i * c_i$  的最大值。

## 输入格式

多组询问, 第一行一个数  $T$  表示数据组数。

对于每组数据, 第一行一个整数  $k$ 。

接下来一行  $2^k - 1$  个数, 表示  $a_1, a_2, \dots, a_{2^k-1}$ 。

接下来一行  $k$  个数, 表示  $b_0, b_1, \dots, b_{k-1}$ 。

## 输出格式

对于每组询问，输出一个数，表示答案。

## 样例输入

```
3
2
1 2 4
4 5
3
3226252 19270256 2430652 1187613 12496062 10286165 17494834
24 85 34
4
901133 6693218 13245489 14740234 16186210 11382783 19277321 3855635 16523184
10168517 16195031 971041 10303441 8395899 11618555
529321239 218214127 92942310 207467810
```

## 样例输出

```
18
1949753378
7832404777567179
```

## 数据范围

对于 10% 的数据， $k \leq 2$ 。  
对于 30% 的数据， $k \leq 3$ 。  
对于另外 20% 的数据， $a_i, b_i$  随机生成。  
对于 100% 的数据， $T \leq 100, 2 \leq k \leq 4, 0 \leq a_i \leq 10^8, 1 \leq b_i \leq 10^9$ 。

## 树 (tree)

### 【题目描述】

你需要维护一棵根为 1 号节点的树。最开始只有树根存在，你需要支持两种操作：

- 加入一个新的节点；
- 查询某个子树的自同构的数量。

其中一棵根为  $r$  的树  $T$  的自同构定义为所有的双射  $f : T \rightarrow T$ ，满足

- $f(r) = r$ ；
- $f(v)$  为  $f(u)$  的父亲当且仅当  $v$  为  $u$  的父亲。

### 【输入格式】

从文件 `tree.in` 中读入数据。  
第一行一个正整数  $m$ ，表示操作的个数。  
接下来  $m$  行，每行两个非负整数 `opt, u`。

- 当  $\text{opt} = 0$  时, 向树中加入一个新的结点. 其编号为树中结点编号最大值  $+1$ , 父亲为  $u$ ;
- 当  $\text{opt} = 1$  时, 查询以  $u$  为根的子树的自同构个数.

## 【输出格式】

输出到文件 `tree.out` 中.

对每个询问操作, 输出一行一个整数, 表示答案对  $998244353$  取模后的值.

## 【样例 1 输入】

```
9
0 1
0 1
1 1
0 2
0 2
0 3
0 3
1 1
1 2
```

## 【样例 1 输出】

```
2
8
2
```

## 【样例 1 解释】

样例给出的即为完全二叉树.

- 第一次查询, 此时树上仅有三个点  $1, 2, 3$ , 其中  $2, 3$  两个点可以对换, 共有  $2$  种自同构.
- 第二次查询, 此时  $2, 3$  两个点,  $4, 5$  两个点,  $6, 7$  两个点均可对换, 共有  $2^3 = 8$  种自同构.
- 第三次查询的子树和第一次查询的树同构, 故答案相同.

## 【样例 2 输入】

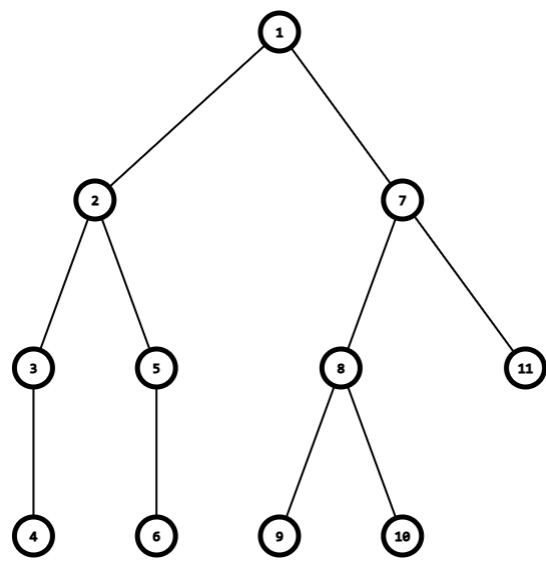
```
11
0 1
0 2
0 3
0 2
0 5
0 1
0 7
0 8
0 8
0 7
1 1
```

【样例 2 输出】

4

【样例 2 解释】

样例中的树如下图



此时 3, 5 两个点和 9, 10 两个点均可对换, 于是共有  $2^2 = 4$  种自同构.

【样例 3】

见选手目录下的 `tree/tree3.in` 与 `tree/tree3.ans`.

该样例约束与子任务 1 一致。

【样例 4】

见选手目录下的 `tree/tree4.in` 与 `tree/tree4.ans`.

该样例约束与子任务 4 一致。

【样例 5】

见选手目录下的 `tree/tree5.in` 与 `tree/tree5.ans`.

该样例约束与子任务 6 一致。

【测试点约束】

对于所有测试数据,  $1 \leq m \leq 3 \times 10^5$ .

本题评测开启捆绑测试.

子任务编号	分数		特殊性质	子任务依赖
1	20	5000	无	无

子任务编号	分数	$m \leq$	特殊性质	子任务依赖
2	5	$3 \times 10^5$	A	无
3	5	$3 \times 10^5$	B	无
4	20	$3 \times 10^5$	C	无
5	20	$2 \times 10^5$	无	1
6	30	$3 \times 10^5$	无	2, 3, 4, 5

特殊性质 A: 保证树的形态是链且 1 为链的一端;

特殊性质 B: 保证树的形态是菊花且 1 为菊花的中心;

特殊性质 C: 树的生成方式为, 加入一个点时, 随机一个合法父亲.