

实外 CCF CSP2022-S 国庆模拟赛

（提高组：第1场）

时间：2022 年 10 月 4 日 14:00 ~ 18:00

题目名称	盘子	数列	点名	树上的数
题目类型	传统型	传统型	传统型	传统型
目录	disk	sequence	rollcall	tree
可执行文件名	disk	sequence	rollcall	tree
输入文件名	disk.in	sequence.in	rollcall.in	tree.in
输出文件名	disk.out	sequence.out	rollcall.out	tree.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	3.0 秒
内存限制	128 MB	512 MB	128 MB	256 MB
子任务数目	5	20	10	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	disk.cpp	sequence.cpp	rollcall.cpp	tree.cpp
对于 C 语言	disk.c	sequence.c	rollcall.c	tree.c
对于 Pascal 语言	disk.pas	sequence.pas	rollcall.pas	tree.pas

编译选项

对于 C++ 语言	-O2
对于 C 语言	-O2
对于 Pascal 语言	-O2

注意事项（请仔细阅读）

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考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 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）。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 全国统一评测时采用的机器配置为：Inter(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
8. 只提供 Linux 格式附加样例文件。
9. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

盘子 (disk)

【题目描述】

有 n 个盘子。盘子被生产出来后，被按照某种顺序摞在一起。初始盘堆中如果一个盘子比所有它上面的盘子都大，那么它是安全的，否则它是危险的。称初始盘堆为A，另外有一个开始为空的盘堆 B。为了掩盖失误，生产商会对盘子序列做一些“处理”，每次进行以下操作中的一个：

- (1) 将 A 最上面的盘子放到 B 最上面；
- (2) 将 B 最上面的盘子给你。

在得到所有 n 个盘子之后，你需要判断初始盘堆里是否有危险的盘子。

【输入格式】

从文件 **disk.in** 中读入数据。

输入文件包含多组数据（不超过 10 组）；

每组数据的第一行为一个整数 n ；

接下来 n 个整数，第 i 个整数表示你收到的第 i 个盘子的大小。

【输出格式】

输出到文件 **disk.out** 中。

对于每组数据，如果不存在安全的初始盘子序列，输出” J”，否则输出” Y”。

【样例输入】

```
3
2 1 3
3
3 1 2
```

【样例输出】

```
Y
J
```

【数据范围】

对于 20% 的数据， $1 \leq n \leq 8$ 。

对于 80% 的数据， $1 \leq n \leq 1000$ 。

对于 100% 的数据， $1 \leq n \leq 100000$ ， $1 \leq \text{盘子大小} \leq 1,000,000,000$ 且互不相等。

数列 (sequence)

【题目描述】

给定整数 n, m, k , 和一个长度为 $m + 1$ 的正整数数组 v_0, v_1, \dots, v_m 。

对于一个长度为 n , 下标从 1 开始且每个元素均不超过 m 的非负整数序列 $\{a_i\}$, 我们定义它的权值为 $v_{a_1} \times v_{a_2} \times \dots \times v_{a_n}$ 。

当这样的序列 $\{a_i\}$ 满足整数 $S = 2^{a_1} + 2^{a_2} + \dots + 2^{a_n}$ 的二进制表示中 1 的个数不超过 k 时, 我们认为 $\{a_i\}$ 是一个合法序列。

计算所有合法序列 $\{a_i\}$ 的权值和对 998244353 取模的结果。

【输入格式】

从文件 *sequence.in* 中读入数据。

输入的一行是三个整数 n, m, k 。

第二行 $m + 1$ 个整数, 分别是 v_0, v_1, \dots, v_m 。

【输出格式】

输出到文件 *sequence.out* 中。

仅一行一个整数, 表示所有合法序列的权值和对 998244353 取模的结果。

【样例输入】

```
1 5 1 1
2 2 1
```

【样例输出】

```
1 40
```

【样例 1 解释】

由于 $k = 1$, 而且由 $n \leq S \leq n \times 2^m$ 知道 $5 \leq S \leq 10$, 合法的 S 只有一种可能: $S = 8$, 这要求 a 中必须有 2 个 0 和 3 个 1, 于是有 $\binom{5}{2} = 10$ 种可能的序列, 每种序列的贡献都是 $v_0^2 v_1^3 = 4$, 权值和为 $10 \times 4 = 40$ 。

【样例 2】

见选手目录下的 *sequence/sequence2.in* 与 *sequence/sequence2.ans*。

【数据范围】

对所有测试点保证 $1 \leq k \leq n \leq 30, 0 \leq m \leq 100, 1 \leq v_i < 998244353$ 。

测试点	n	k	m
1 ~ 4	= 8	$\leq n$	= 9
5 ~ 7	= 30		= 7
8 ~ 10			= 12
11 ~ 13		= 1	= 100
14 ~ 15	= 5	$\leq n$	= 50
16	= 15		= 100
17 ~ 18	= 30		= 30
19 ~ 20			= 100

点名 (rollcall)

【题目描述】

在 A 班的体育课上，同学们常常会迟到几分钟，但体育老师的点名却一直很准时。老师只关心同学的身高，他会依次询问当前最矮的身高，次矮的身高，第三矮的身高，等等。

在询问的过程中，会不时地有人插进队伍里。

你需要回答老师每次的询问。

【输入格式】

从文件 **rollcall.in** 中读入数据。

第一行两个整数 n m ，表示先后有 n 个人进队，老师询问了 m 次；

第二行 n 个整数，第 i 个数 A_i 表示第 i 个进入队伍的同学的身高为 A_i ；

第三行 m 个整数，第 j 个数 B_j 表示老师在第 B_j 个同学进入队伍后有一次询问。

【输出格式】

输出到文件 **rollcall.out** 中。

m 行，每行一个整数，依次表示老师每次询问的答案。数据保证合法。

【样例输入】

```
7 4
9 7 2 8 14 1 8
1 2 6 6
```

【样例输出】

```
9
9
7
8
```

【样例解释】

(9){No.1 = 9}; (9 7){No.2 = 9}; (9 7 2 8 14 1){No.3 = 7; No.4 = 8}

【数据范围】

对于 40% 的数据， $1 \leq n \leq 1000$ 。

对于 100% 的数据， $1 \leq m \leq n \leq 30000$ ， $0 \leq A_i \leq 2^{32}$ 。

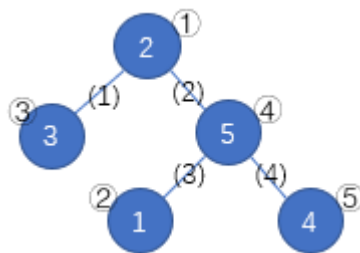
树上的数 (tree)

【题目描述】

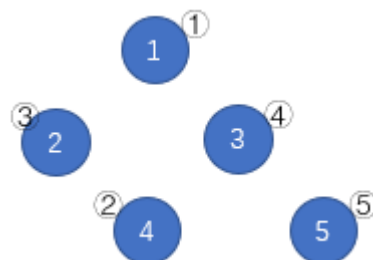
给定一个大小为 n 的树，它共有 n 个结点与 $n-1$ 条边，结点从 $1 \sim n$ 编号。初始时每个结点上都有一个 $1 \sim n$ 的数字，且每个 $1 \sim n$ 的数字都只在恰好一个结点上出现。

接下来你需要进行恰好 $n-1$ 次删边操作，每次操作你需要选一条未被删去的边，此时这条边所连接的两个结点上的数字将会交换，然后这条边将被删去。

$n-1$ 次操作过后，所有的边都将被删去。此时，按数字从小到大的顺序，将数字 $1 \sim n$ 所在的结点编号依次排列，就得到一个结点编号的排列 P_i 。现在请你求出，在最优操作方案下能得到的字典序最小的 P_i 。



如上图，蓝圈中的数字 $1 \sim 5$ 一开始分别在结点 ②、①、③、⑤、④。按照 (1)(4)(3)(2) 的顺序删去所有边，树变为下图。按数字顺序得到的结点编号排列为 ①③④②⑤，该排列是所有可能的结果中字典序最小的。



【输入格式】

从文件 `tree.in` 中读入数据。

本题输入包含多组测试数据。

第一行一个正整数 T ，表示数据组数。

对于每组测试数据：

第一行一个整数 n ，表示树的大小。

第二行 n 个整数，第 i ($1 \leq i \leq n$) 个整数表示数字 i 初始时所在的结点编号。

接下来 $n-1$ 行每行两个整数 x, y ，表示一条连接 x 号结点与 y 号结点的边。

【输出格式】

输出到文件 *tree.out* 中。

对于每组测试数据，输出一行共 n 个用空格隔开的整数，表示最优操作方案下所能得到的字典序最小的 P_i 。

【样例 1 输入】

```
4
5
2 1 3 5 4
1 3
1 4
2 4
4 5
5
3 4 2 1 5
1 2
2 3
3 4
4 5
5
1 2 5 3 4
1 2
1 3
1 4
1 5
10
1 2 3 4 5 7 8 9 10 6
1 2
1 3
1 4
1 5
5 6
6 7
7 8
8 9
9 10
```

【样例 1 输出】

1 3 4 2 5
1 3 5 2 4
2 3 1 4 5
2 3 4 5 6 1 7 8 9 10

【样例 2】

见选手目录下的 *tree/tree2.in* 与 *tree/tree2.ans*。

【数据范围】

测试点编号	$n \leq$	特殊性质
1 ~ 2	10	无
3 ~ 4	160	树的形态是一条链
5 ~ 7	2000	
8 ~ 9	160	存在度数为 $n - 1$ 的结点
10 ~ 12	2000	
13 ~ 16	160	无
17 ~ 20	2000	

对于所有测试点： $1 \leq T \leq 10$ ，保证给出的是一个树。