

# NOI2022 省选

## DAY2

时间：2022 年 4 月 17 日 08:30 ~ 13:00

题目名称	卡牌	序列变换	最大权独立集问题
题目类型	传统型	传统型	传统型
目录	card	bracket	mis
可执行文件名	card	bracket	mis
输入文件名	card.in	bracket.in	mis.in
输出文件名	card.out	bracket.out	mis.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	2048 MiB
测试点数目	20	25	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	card.cpp	bracket.cpp	mis.cpp
-----------	----------	-------------	---------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

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. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 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。
9. 只提供 Linux 格式附加样例文件。
10. 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。

## 卡牌 (card)

### 【题目描述】

小 A 有  $n$  张卡牌，编号为  $1, 2, \dots, n$ 。每张卡牌上写着一个正整数，第  $i$  张卡牌上的正整数为  $s_i$ 。

现在有  $m$  轮游戏，第  $i$  轮游戏会给出  $c_i$  个质数，小 A 需要选择任意多张卡牌，使得这些卡牌上面的正整数的乘积能被该轮游戏给出的每个质数整除。

这当然难不倒小 A，于是他开始思考一个更难的问题，对于每一轮游戏，他有多少种卡牌的选法。

这给小 A 整不会了，于是他只能来求助你，你只需要告诉他答案模 998244353 的值即可。两种选法 A 和 B 互不相同当且仅当存在一张卡牌在 A 中被选择但在 B 中未被选择或者存在一张卡牌在 B 中被选择但在 A 中未被选择。注意：牌面上的数字相同但编号不相同的两张卡牌被视为不同的卡牌。

### 【输入格式】

从文件 `card.in` 中读入数据。

第一行一个正整数  $n$ ，表示卡牌数量。

第二行  $n$  个正整数  $s_i$ ，表示每张卡牌上写的数字。

第三行一个正整数  $m$ ，表示游戏轮数。

接下来  $m$  行，每行第一个正整数  $c_i$ ，表示该轮游戏给出的质数个数，接下来  $c_i$  个质数  $p_{i,j}$ ，表示该轮游戏给出的所有质数。数据保证  $\sum_i c_i \leq 18000$ ，即所有  $c_i$  之和不超过 18000。

### 【输出格式】

输出到文件 `card.out` 中。

输出  $m$  行，每行一个整数，第  $i$  行表示第  $i$  轮游戏的方案数模 998244353 的值。

### 【样例 1 输入】

```
1 5
2 10 2 10 5 46
3 4
4 2 2 5
5 2 2 23
6 1 3
7 1 23
```

## 【样例 1 输出】

```

1 27
2 16
3 0
4 16

```

## 【样例 1 解释】

第一轮游戏：除了以下 5 种方案外其它方案都可行：什么都不选、选 2、选 5、选 46、选 2 和 46。所以答案为  $2^5 - 5 = 27$

第二轮游戏：只要选了 46，其它卡牌选不选均可，所以答案为  $2^4 = 16$ 。

## 【样例 2】

见选手目录下的 *card/card2.in* 与 *card/card2.ans*。

## 【数据范围】

对于 100% 的数据， $n \leq 10^6, s_i \leq 2000, m \leq 1500, \sum_i c_i \leq 18000, 2 \leq p_{i,j} \leq 2000$

测试点	$n \leq$	$m \leq$	$\sum_i c_i \leq$	其他限制
1,2	10	10	20	$s_i \leq 30$
3 ~ 5		20	50	无
6 ~ 8	$10^6$	1500	10000	$s_i \leq 30$
9 ~ 11	10000	1000	5000	$s_i \leq 500$
12,13	1000	100	1000	无
14 ~ 17	5000	600	7000	
18 ~ 20	$10^6$	1500	18000	

## 序列变换 (bracket)

### 【题目描述】

你手里有一个长度为  $2n$  的合法括号序列  $s$ 。 $s$  的每一个左括号有一个权值。

在你眼中,不同的括号序列带来的视觉美感不尽相同。因此,你对具有某一种结构的括号序列特别喜欢,而讨厌具有其他一些结构的括号序列。你希望对  $s$  进行一些变换,以消除掉一些自己不喜欢的结构。

具体而言,形如  $(A()B)$  (其中  $A,B$  均为合法括号序列,下同)的结构是你喜欢的,而形如  $(A)(B)$  的结构是你不喜欢的。你有两种操作来改变括号之间的位置。

这两种操作如下:

- 操作 1: 交换形如  $p(A)(B)q$  的串中  $A$  和  $B$  之间的两个括号,变换为  $p(A()B)q$  (其中  $p,q$  为任意串,可以为空,但不一定分别为合法括号序列,下同),它的代价为  $x$  乘  $(A)$  中第一个左括号的权值加上  $y$  乘  $(B)$  中第一个左括号的权值,其中  $x,y \in \{0,1\}$ ;
- 操作 2: 交换形如  $pABq$  的串中的  $A$  和  $B$ ,变换为  $pBAq$ ,这个操作不需要代价。注意:交换的时候所有左括号的权值是跟着这个括号一起交换的。

你现在想知道的是,将  $s$  变换为一个不包含你不喜欢的结构的括号序列至少需要多少代价?

### 【输入格式】

从文件 `bracket.in` 中读入数据。

第一行三个整数  $n, x, y$ 。

第二行一个长度为  $2n$  的合法括号序列,表示  $s$ 。

第三行  $n$  个正整数,其中第  $i$  个表示左数第  $i$  个左括号的权值。

### 【输出格式】

输出到文件 `bracket.out` 中。

一行一个整数,表示把  $s$  变换为一个不包含你不喜欢的结构的括号序列至少需要的代价。

### 【样例 1 输入】

```
1 2 0 1
2 ()()
3 1 3
```

**【样例 1 输出】**

```
1 1
```

**【样例 1 解释】**

最优方案是先使用操作 2 交换两对括号，然后使用操作 1（此时 **A**, **B**, **p**, **q** 都是空串）交换中间的两个括号，代价为 **B** 左边那个括号的权值，也就是 1。最后得到括号序列  $(( ))$ ，不包含你不喜欢的结构。

**【样例 2 输入】**

```
1 2 1 0
2 ()()
3 1 3
```

**【样例 2 输出】**

```
1 1
```

**【样例 2 解释】**

最优方案是直接使用操作 1，因为此时计算代价的方式不同了，这次只算 **A** 左边的那个括号的权值作为代价。

**【样例 3】**

见选手目录下的 *bracket/bracket3.in* 与 *bracket/bracket3.ans*。

**【子任务】**

保证  $2 \leq n \leq 400000$ ,  $0 \leq x, y \leq 1$ 。

保证所有的权值在  $[1, 10^7]$  之内。

测试点编号	特殊限制
1 ~ 3	$n \leq 8$
4 ~ 5	所有权值均相等
6 ~ 8	$n \leq 20$
9 ~ 12	$x = 0, y = 1$
13 ~ 16	$n \leq 2000$
17 ~ 25	无特殊限制

【提示】

称一个字符串  $s$  为合法括号序列，当且仅当  $s$  仅由数量相等的字符 ( 和 ) 组成，且对于  $s$  的每一个前缀而言，其中 ( 的数量均不少于 ) 的数量。特别地，空串也是合法括号序列。

## 最大权独立集问题 (mis)

### 【题目描述】

小 E 喜欢出最大权独立集问题。

接下来，他还想了  $n$  道最大权独立集问题。

小 E 有  $n$  个 AI，编号为  $1 \sim n$ 。

开始时第  $i$  个 AI 里面存有  $d_i$  道小 E 事先出好的最大权独立集问题。

有些 AI 之间可以互相通信，对于所有的  $2 \leq i \leq n$ ，第  $i$  个 AI 可以和第  $c_i$  个 AI 互相通信。其中  $c_i < i$ ，且相同的  $c_i$  出现不超过 2 次。所以，这些 AI 连成了一个二叉树的形状。此外，其他对 AI 不可以互相通信。

小 E 需要暂时断开这些 AI 之间的连接。他只能逐一断开 AI 之间的连接。两个原本能够互相通信的 AI 在断开它们之间的连接之前，会互相交换存在里面的所有题目，具体请见样例。

小 E 希望在断掉所有连接之后，参与交换的题目数量最少。

他想叫你帮他解决这个问题，还说如果你成功解决了这个问题，那么在出那些最大权独立集问题的时候，他会帮你提交一份标程代码。

### 【输入格式】

从文件 `mis.in` 中读入数据。

第一行一个正整数  $n$ 。

第二行  $n$  个整数，第  $i$  个表示  $d_i$ 。

第三行  $n - 1$  个正整数，第  $i$  个表示  $c_{i+1}$ 。

### 【输出格式】

输出到文件 `mis.out` 中。

一行一个整数表示答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 7
```

【样例 1 解释】

一种最优的方案是：断开 1 号与 2 号 AI 之间的连接，这样需要交换  $2 + 1 = 3$  道题；然后断开 1 号与 3 号 AI 之间的连接，这样需要交换  $1 + 3 = 4$  道题。所以答案是 7。

【子任务】

保证  $1 \leq c_i < i$ ，且相同的  $c_i$  最多出现 2 次。  
保证  $1 \leq d_i \leq 10^9$ 。

测试点编号	$n \leq$
1 ~ 3	10
4 ~ 7	100
8 ~ 11	500
12 ~ 16	1000
17 ~ 25	5000