全国青少年信息学奥林匹克联赛模拟赛

NPIO Round 6

时间：2024 年 11 月 13 日 07:30 ~ 11:50

题目名称	数的拆分	括号序列	数据结构	构造数组
题目类型	传统型	传统型	传统型	传统型
目录	`number`	`bracket`	`struct`	`array`
可执行文件名	`number`	`bracket`	`struct`	`array`
输入文件名	`number.in`	`bracket.in`	`struct.in`	`array.in`
输出文件名	`number.out`	`bracket.out`	`struct.out`	`array.out`
每个测试点时限	4.0 秒	2.0 秒	2.0 秒	4.0 秒
内存限制	1 GiB	1 GiB	1 GiB	1 GiB
是否打包	是	是	否	否
子任务或测试点数目	9	10	20	25
测试点是否等分	否	否	是	是

提交源程序文件名

对于 C++ 语言	`number.cpp`	`bracket.cpp`	`struct.cpp`	`array.cpp`

编译选项

对于 C++ 语言	`-O2 -std=c++14 -static`

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. `main` 函数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. 提交的程序的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，结果比较方式为全文比较（忽略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. 统一评测时采用的机器配置为：Intel(R) Core(TM) i5-6500 CPU @3.20GHz，内存 8GB。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。

数的拆分 (number)

【题目描述】

给定 T 个正整数 a_i，分别问每个 a_i 能否表示为 $x_1^{y_1} \cdot x_2^{y_2}$ 的形式，其中 x_1, x_2 为正整数，y_1, y_2 为大于等于 2 的正整数。

【输入格式】

从文件 ***number.in*** 中读入数据。
输入第一行包含一个整数 T 表示询问次数。
接下来 T 行，每行包含一个正整数 a_i 。

【输出格式】

输出到文件 ***number.out*** 中。
对于每次询问，如果 a_i 能够表示为题目描述的形式则输出 `yes`，否则输出 `no`。

【样例 1 输入】

```
7
2
6
12
4
8
24
72
```

【样例 1 输出】

```
no
no
no
yes
yes
no
yes
```

【样例 1 解释】

第 $4,5,7$ 个数分别可以表示为:

$$a_4 = 2^2 \times 1^2;$$
$$a_5 = 2^3 \times 1^2;$$
$$a_7 = 2^3 \times 3^2.$$

【样例 2】

见选手目录下的 ***number/number2.in*** 与 ***number/number2.ans***。
该样例满足子任务 2 的条件限制。

【样例 3】

见选手目录下的 ***number/number3.in*** 与 ***number/number3.ans***。
该样例满足子任务 4 的条件限制。

【样例 4】

见选手目录下的 ***number/number4.in*** 与 ***number/number4.ans***。
该样例满足子任务 6 的条件限制。

【样例 5】

见选手目录下的 ***number/number5.in*** 与 ***number/number5.ans***。
该样例满足子任务 8 的条件限制。

【数据范围】

对于所有测试数据保证：$1 \le T \le 4 \times 10^5, 1 \le a_i \le 10^{18}$。

子任务编号	$T \le$	$a_i \le$	得分
1	5	5	10
2	10	1000	10
3	1000	10^4	15
4	10^5	10^6	10
5		10^7	10
6	1000	10^9	15
7	500	10^{16}	10
8	4×10^5	10^{13}	15
9		10^{18}	5

括号序列 (bracket)

【题目描述】

给定一个长度为 n 的括号序列和一个长度为 n 的排列 P。一个括号序列被称为**好**的，当且仅当：

- 这个括号序列是**合法**的；
- 构造一张 n 个点的图，当且仅当第 i 个位置是左括号时，在点 i 与点 P_i 间连一条无向边，最后形成的图满足每个点的度数（与这个点相连的边数）均为一。

一个括号序列**合法**的定义如下：

- 空序列是合法的；
- 如果 A 是合法的，那么 (A) 也是合法的；
- 如果 A 和 B 都是合法的，那么 AB 也是合法的。

例如 ()()((()())) 是合法的，而 ())(() 不是。
现在请你给出一种**好**的括号序列。保证在给定数据下存在至少一种**好**的括号序列。

【输入格式】

从文件 ***bracket.in*** 中读入数据。
第一行一个整数 n，表示括号序列的长度。
接下来一行 n 个正整数，第 i 个数表示 P_i。

【输出格式】

输出到文件 ***bracket.out*** 中。
输出一行一个长度为 n 的括号序列，如果有多种解，输出任意一种即可。
注意，样例输出只是一种参考解，解可能并不唯一。

【样例 1 输入】

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

【样例 1 输出】

```
()()()
```

【样例 2】

见选手目录下的 ***bracket/bracket2.in*** 与 ***bracket/bracket2.ans***。
该样例满足子任务 2 的条件限制。

【样例 3】

见选手目录下的 ***bracket/bracket3.in*** 与 ***bracket/bracket3.ans***。
该样例满足子任务 4 的条件限制。

【样例 4】

见选手目录下的 ***bracket/bracket4.in*** 与 ***bracket/bracket4.ans***。
该样例满足子任务 8 的条件限制。

【样例 5】

见选手目录下的 ***bracket/bracket5.in*** 与 ***bracket/bracket5.ans***。
该样例满足子任务 10 的条件限制。

【数据范围】

对于所有测试数据保证：$2 \le n \le 100, 1 \le P_i \le n, 2|n$ 且 P_i 是一个排列，对于任意 i 有 $i \neq P_i$。

本题采用打包评测，并按照逻辑关系绑定依赖，各子任务的附加限制如下表所示：

子任务编号	$n \le$	特殊性质	得分
1	4	无	5
2	20		10
3	24		15
4	28		10
5	32		15
6	40	A	5
7		B	10
8		无	10
9	70		10
10	100		10

特殊性质 A：$P_i = ((i+1) \operatorname{xor} 1) - 1$，其中 xor 表示按位异或运算。

特殊性质 B：$P_i = \begin{cases} i + \frac{n}{2} & , i \le \frac{n}{2} \\ (i \bmod \frac{n}{2}) + 1 & , i > \frac{n}{2} \end{cases}$。

【校验器】

为了方便选手测试，在附件中的 ***bracket*** 目录下我们下发了 ***checker.cpp*** 文件，选手可以编译该程序，并使用它校验自己的输出文件。但请注意它与最终评测时所使用的校验器并不完全一致。你也不需要关心其代码的具体内容。

编译命令为：`g++ checker.cpp -o checker -std=c++14`

`checker` 的使用方式为：`checker <inputfile> <outputfile> <answerfile>`，参数依次表示输入文件、你的输出文件以及答案文件。

若你输出的括号序列不合法或者图的度数不符合条件，则校验器会给出相应提示。

若你的方案正确，校验器会给出 `OK`。

【温馨提示】

如果你认为你的做法是正解，请相信它的常数足以通过。

数据结构 (struct)

【题目描述】

给定一个长度为 n 的序列 a，序列的标号从 1 开始，每一个位置是一个 $[1,n]$ 内的整数。

有 m 个操作，第 i 次操作给出一个区间 $[l_i,r_i]$：这次操作会先将下标在 $[l_i,r_i]$ 中的所有元素加上 1，然后询问全局颜色数，并在最后撤销这次修改（也就是说操作之间互不影响）。

其中，一个序列的全局颜色数定义为这个序列中出现的不同数值的个数。

【输入格式】

从文件 ***struct.in*** 中读入数据。

第一行两个整数 n,m。

第二行 n 个用空格隔开的数表示序列 a。

之后 m 行，每行两个用空格隔开的数 l_i,r_i 表示一次询问。

【输出格式】

输出到文件 ***struct.out*** 中。

对每次操作，输出一行一个数表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
6
7
6
9
8
9
8
8
8
```

【样例 2】

见选手目录下的 ***struct/struct2.in*** 与 ***struct/struct2.ans***。
该样例满足测试点 $1 \sim 2$ 的条件限制。

【样例 3】

见选手目录下的 ***struct/struct3.in*** 与 ***struct/struct3.ans***。
该样例满足测试点 $6 \sim 7$ 的条件限制。

【样例 4】

见选手目录下的 ***struct/struct4.in*** 与 ***struct/struct4.ans***。
该样例满足测试点 $10 \sim 13$ 的条件限制。

【样例 5】

见选手目录下的 ***struct/struct5.in*** 与 ***struct/struct5.ans***。
该样例满足测试点 $14 \sim 16$ 的条件限制。

【样例 6】

见选手目录下的 ***struct/struct6.in*** 与 ***struct/struct6.ans***。
该样例满足测试点 $17 \sim 20$ 的条件限制。

【数据范围】

对于所有数据保证：$1 \leq n, m \leq 10^6, 1 \leq a_i \leq n, 1 \leq l_i \leq r_i \leq n$。
$\forall 1 \leq i < m, r_i \leq r_{i+1}$。

测试点编号	$n \leq$	$m \leq$	特殊性质
$1 \sim 2$	5000	5000	无
$3 \sim 5$		10^6	
$6 \sim 7$	5×10^4	5×10^4	
$8 \sim 9$	2×10^5	2×10^5	
$10 \sim 13$	10^6	10^6	A
$14 \sim 16$			B
$17 \sim 20$			无

特殊性质 A：$\forall 1 \leq i \leq n, 2 \nmid a_i$。

特殊性质 B：$\forall 1 \leq i < j \leq n \wedge a_i = a_j = c, \forall i \leq k \leq j, a_k = c$。

构造数组 (array)

【题目描述】

你现在有一个长度为 n 的数组 a。一开始，所有 a_i 均为 0。给出一个同样长度为 n 的目标数组 b。求有多少种方案，使得通过若干次以下操作，可以让 a 数组变成 b。

- 选出两个**不同的**下标 $1 \le i < j \le n$，并将 a_i 和 a_j 同时增加 1。

两种方案被称之为不同的，当且仅当存在一个 x 使得一种方案中第 x 次操作选择的两个下标 (i, j) 与另一种方案中的不同。

答案对 998244353 取模。

【输入格式】

从文件 ***array.in*** 中读入数据。

输入数据一共包含两行。

第一行包含一个正整数 n。

第二行包含 n 个正整数，表示 $b_1, b_2, \ldots, b_n$。

【输出格式】

输出到文件 ***array.out*** 中。

输出一行一个正整数,表示将 a 数组通过若干次操作变成 b 数组的方案数对 998244353 取模后的结果。

【样例 1 输入】

```
4
2 2 2 2
```

【样例 1 输出】

```
90
```

【样例 1 解释】

种类编号	第一组	第二组	第三组	第四组	方案数
1	1,2	1,2	3,4	3,4	$\binom{4}{2}=6$
2	1,3	1,3	2,4	2,4	$\binom{4}{2}=6$
3	1,4	1,4	2,3	2,3	$\binom{4}{2}=6$
4	1,2	1,4	2,3	3,4	$4!=24$
5	1,2	1,3	2,4	3,4	$4!=24$
6	1,3	1,4	2,3	2,4	$4!=24$

总方案数是 $6\times 3+24\times 3=90$。

【样例 2】

见选手目录下的 ***array/array2.in*** 与 ***array/array2.ans***。
该样例满足测试点 6 ∼ 8 的条件限制。

【样例 3】

见选手目录下的 ***array/array3.in*** 与 ***array/array3.ans***。
该样例满足测试点 12 ∼ 14 的条件限制。

【样例 4】

见选手目录下的 ***array/array4.in*** 与 ***array/array4.ans***。
该样例满足测试点 15 ∼ 18 的条件限制。

【样例 5】

见选手目录下的 ***array/array5.in*** 与 ***array/array5.ans***。
该样例满足测试点 19 ∼ 20 的条件限制。

【样例 6】

见选手目录下的 ***array/array6.in*** 与 ***array/array6.ans***。
该样例满足测试点 21 ∼ 22 的条件限制。

【样例 7】

见选手目录下的 ***array/array7.in*** 与 ***array/array7.ans***。
该样例满足测试点 23 ∼ 25 的条件限制。

【数据范围】

对于所有数据，保证 $1 \le n \le 5000$，$1 \le b_i \le 30000$，$\sum b_i \le 30000$。

测试点编号	n	$\sum b_i$
1	≤ 5000	$\equiv 1 \pmod 2$
$2 \sim 3$	$= 1$	≤ 30000
$4 \sim 5$	$= 2$	
$6 \sim 8$	≤ 5	≤ 8
$9 \sim 11$	≤ 20	$= n$
$12 \sim 14$	≤ 5000	
$15 \sim 18$	≤ 16	≤ 16
$19 \sim 20$	≤ 700	≤ 700
$21 \sim 22$	≤ 5000	≤ 5000
$23 \sim 25$		≤ 30000