NOIP 2022 模拟赛

KDOI Round 3

时间：2022 年 11 月 20 日 08:30 ~ 13:00

题目名称	还原数据	构造数组	I LOVE U TOO	序列变换
题目类型	传统型	传统型	传统型	传统型
可执行文件名	`restore`	`array`	`ilu`	`control`
时间限制	1.0 秒	4.0 秒	2.0 秒	4.0 秒
空间限制	512 MiB	512 MiB	512 MiB	1.0 GiB
测试点数目	10	25	10	25
测试点是否等分	是	是	是	是

提交源文件程序名

对于 C++ 语言	`restore.cpp`	`array.cpp`	`ilu.cpp`	`control.cpp`

编译选项

对于 C++ 语言	`-O2 -lm`

注意事项（请仔细阅读）

1. 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。
2. 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）。
3. 选手提交的程序源文件必须不大于 100KB。
4. 程序可使用的栈空间内存限制与题目的内存限制一致。
5. 只提供 Linux 格式附加样例文件。
6. 禁止在源代码中改变编译器参数（如使用 #pragma 命令），禁止使用系统结构相关指令（如内联汇编）和其他可能造成不公平的方法。
7. **所有题目均使用标准输入输出。**

还原数据（restore）

【题目描述】

小 E 正在做一道经典题：

给定一个长度为 n 的序列 a 和 q 个操作，操作共有 2 种类型：

- `1 l r x`：对于所有 $l \le i \le r$，$a_i \leftarrow a_i + x$。
- `2 l r x`：对于所有 $l \le i \le r$，$a_i \leftarrow \max(a_i, x)$。

题目要求输出所有操作结束后的最终序列 a'。

小 E 迅速写了一份代码提交，但是发现，由于宇宙射线的影响，输入数据出现了一些小问题。具体地，对于所有 2 操作，操作中给出的 x 均被丢失了，也就是说，输入数据中的 2 操作只剩下了 `2 l r`。输出数据则没有问题。小 E 现在想要通过剩余的数据恢复原来的输入数据，请你帮助他完成这个任务。

当然，可能会有多种合法的输入数据，你需要找到其中任意一种。数据保证有解。

【输入格式】

从标准输入读入数据。

本题有多组测试数据。

第一行一个正整数 T，表示数据组数。

对于每组测试数据，第一行两个非负整数 n, q。

第二行 n 个整数，表示初始序列 $a_1, a_2, \cdots, a_n$。

接下来 q 行，每行一次操作，形如 `1 l r x` 或 `2 l r`。

接下来一行 n 个整数，表示最终序列 $a'_1, a'_2, \cdots, a'_n$。

【输出格式】

输出到标准输出。

本题开启自定义校验器（Special Judge）。

对于每组测试数据，设共有 q_2 个 2 操作，输出一行 q_2 个整数，第 i 个整数表示第 i 次 2 操作中所给出的 x 的值。

你需要保证 $-10^{15} \le x \le 10^{15}$。

【样例 1 输入】

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

```
2 3 5
1 3 4 2
2 1 1
20 2 5 6 5
```

【样例 1 输出】

```
3 20
```

【样例 1 解释】

所有合法输出需要满足：第 1 个数 ≤ 3，第 2 个数恰好为 20。

【样例 2】

见选手文件中的 ***restore/restore2.in*** 与 ***restore/restore2.ans***。

【样例 3】

见选手文件中的 ***restore/restore3.in*** 与 ***restore/restore3.ans***。

【数据范围】

记 q_2 为单组数据内 2 操作的个数，$\sum n$ 为单个测试点内所有 n 的和，$\sum q$ 为单个测试点内所有 q 的和。

对于 20% 的数据，保证 $n, q \leq 50$，$\sum n, \sum q \leq 1\,000$。

对于 40% 的数据，保证 $n, q \leq 1\,000$，$\sum n, \sum q \leq 10^5$。

对于另外 20% 的数据，保证 $l = 1, r = n$。

对于另外 20% 的数据，保证 $q_2 \leq 100$。

对于 100% 的数据，保证 $1 \leq T \leq 100$，$1 \leq n, q \leq 10^5$，$1 \leq \sum n, \sum q \leq 3 \times 10^5$，$-10^9 \leq a_i, x \leq 10^9$，$-10^{15} \leq a'_i \leq 10^{15}$，$q_2 \geq 1$。

【校验器】

为了方便测试，在 `restore` 目录下我们下发了 `checker.cpp` 文件。你可以编译该文件，并使用它校验自己的输出文件。请注意它与最终评测时所用的校验器并不完全一致，你不需要也不应该关心其代码的具体内容。

编译命令为：

```
g++ checker.cpp -o checker -std=c++14
```

使用方式为：

```
./checker <inputfile> <outputfile> <answerfile>
```

校验器可能会返回以下状态中的其中一种：

- `Accepted`：表示你的输出完全正确。
- `Wrong answer at testcase x`：表示你的输出在第 x 个测试数据出错。

【提示】

本题输入输出量较大，建议使用较快的输入输出方式。

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，当且仅当存在至少一个 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】

从标准输入读入数据。

输入数据一共包含两行。

第一行包含一个正整数 n。

第二行 n 个正整数，表示 $b_1, b_2, \cdots, b_n$。

【输出格式】

输出到标准输出。

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<->3	2<->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】

见选手文件中的 ***array/array2.in*** 与 ***array/array2.ans***。
此样例满足测试点 $6\sim 8$ 的限制。

【样例 3】

见选手文件中的 ***array/array3.in*** 与 ***array/array3.ans***。
此样例满足测试点 $12\sim 14$ 的限制。

【样例 4】

见选手文件中的 ***array/array4.in*** 与 ***array/array4.ans***。
此样例满足测试点 $15\sim 18$ 的限制。

【样例 5】

见选手文件中的 ***array/array5.in*** 与 ***array/array5.ans***。
此样例满足测试点 $19\sim 20$ 的限制。

【样例 6】

见选手文件中的 ***array/array6.in*** 与 ***array/array6.ans***。
此样例满足测试点 $21\sim 22$ 的限制。

【样例 7】

见选手文件中的 ***array/array7.in*** 与 ***array/array7.ans***。
此样例满足测试点 $23\sim 25$ 的限制。

【数据范围】

对于 100% 的数据，$1 \le n \le 5\ 000$，$1 \le b_i \le 30\ 000$，$\sum b_i \le 30\ 000$。

测试点编号	n	$\sum b_i$
1	$\le 5\ 000$	$\equiv 1 \pmod 2$
$2 \sim 3$	$= 1$	$\le 30\ 000$
$4 \sim 5$	$= 2$	
$6 \sim 8$	≤ 5	≤ 8
$9 \sim 11$	≤ 20	$= n$
$12 \sim 14$	$\le 5\ 000$	
$15 \sim 18$	≤ 16	≤ 16
$19 \sim 20$	≤ 700	≤ 700
$21 \sim 22$	$\le 5\ 000$	$\le 5\ 000$
$23 \sim 25$		$\le 30\ 000$

I LOVE U TOO（ilu）

【题目背景】

I LOVE U TOO
「因为有你 我的舞台 才会发亮
化身炙热的太阳 在彼此的世界闪光
欢呼响起 我的声音 还在回响
台下挥舞的荧光 连接成蓝色的海洋」

天依和阿绫是一对好朋友。
一天晚上，天依和阿绫闲着没事干，打算出道题。
阿绫：“我有很多的数。”
天依：“那我要求和！”
阿绫：“再来个最大值咋样？”
天依：“不行，我要最小值！”
阿绫：“最大值，好嘛...”
于是，就有了这道题。

【题目描述】

天依给了你一个数 n，表示现在有一个 n 层的金字塔形的格子纸，第 i 层有 i 个格子。**保证 n 是 2 的正整数次幂。**

阿绫给了你 $\frac{n\times(n+1)}{2}$ 个数，分别为 $1,2,\cdots,\frac{n\times(n+1)}{2}$，现在你要将这些数分别填入这 $\frac{n\times(n+1)}{2}$ 个格子。

定义一条到 i 的路径为从第一行的格子走到最底层从左往右第 i 个格子，每次只能往左下或者右下走的路径。

定义一条路径的权值为其所有经过的格子的权值的和。

定义 $f(i)$ 表示所有到 i 的路径的权值的最大值。

定义一种填放方案的权值为 $\min\limits_{i=1}^{n} f(i)$。现在你要求出所有填放方案的最大权值，并输出任意一种权值最大的填放方案。

【输入格式】

从标准输入读入数据。

输入数据仅包含一行一个正整数 n。

保证 n 是 2 的正整数次幂。

【输出格式】

输出到标准输出。

本题开启自定义校验器（Special Judge）。

第 1 行一个正整数 k，表示所有方案中最大的权值。

第 2 至 $n+1$ 行，第 $i+1$ 行包含 i 个数，表示填放方案。

【样例 1 输入】

```
2
```

【样例 1 输出】

```
4
3
1 2
```

【数据范围】

对于第 i 个测试点，$n=2^{i+1}$，总计 10 个测试点。

对于 100% 的数据，$2 \le n \le 2\ 048$。

【评分方式】

对于每个测试点，如果你输出的 k 和你的填数方案的权值不同，你将在该测试点得到 0 分。

否则，记标准答案为 x，设 $p=\frac{x-k}{x}$，则你的得分由下表给出：

$p \le$	0	0.005	0.06	0.1	0.15	0.3	0.6
分数	10	8	7	6	5	3	1

序列变换（control）

【题目描述】

给定一个长度为 n 的 **01** 序列 a 和 q 次询问，询问参数 k。

每次询问给定 L, R，其中 $1 \le L \le R \le n$，你可以进行如下操作：

- 选择一个下标 $L < i \le R$；
- 将 a_{i-1} 赋值为 $a_{i-1} \oplus a_i$，a_{i+1} 赋值为 $a_{i+1} \oplus a_i$。如果 $i = n$，则不对 a_{i+1} 作出改变。其中 $\oplus$ 表示按位异或运算。

求使得 $[L, R]$ 区间内至多有 k 个 **1** 的最小操作次数。询问之间相互独立，也就是说，每次询问后重置为初始序列。

【输入格式】

从标准输入读入数据。

第一行三个正整数 n, k, q。

第二行 n 个正整数 $a_1, a_2, \cdots, a_n$。

接下来 q 行，每行两个正整数 L, R，表示一次询问。

【输出格式】

输出到标准输出。

输出共 q 行，每行一个整数，表示答案。

【样例 1 输入】

```
5 1 2
1 1 1 0 1
2 3
1 3
```

【样例 1 输出】

```
1
1
```

【样例 1 解释】

如图，用绿色代表 `0`，红色代表 `1`，初始序列如下：

对于第 1 次询问，选择 $i = 3$，则序列变为下图：

对于第 2 次询问，选择 $i = 2$，则序列变为下图：

【样例 2 输入】

```
20 3 22
0 0 1 1 1 1 1 0 0 0 0 0 1 0 1 0 0 1 0 1
12 15
1 6
5 10
2 5
9 18
6 17
2 13
4 16
2 8
9 19
10 15
7 15
1 3
14 18
6 17
12 14
7 16
14 18
11 12
3 5
```

```
3 6
3 15
```

【样例 2 输出】

```
0
1
0
0
0
6
3
5
1
0
0
0
0
0
6
0
0
0
0
0
1
3
```

【样例 2 解释】

对于第 1 次询问，由于 $a_{12}, a_{13}, a_{14}, a_{15}$ 中只有 2 个 `1`，所以不需要进行任何操作。
对于第 6 次询问，可以依次选择 $i = \{7, 8, 9, 10, 11, 12\}$。

【样例 3】

见选手文件中的 ***control/control3.in*** 与 ***control/control3.ans***。
此样例满足测试点 7 ∼ 10 的限制。

【样例 4】

见选手文件中的 ***control/control4.in*** 与 ***control/control4.ans***。
此样例满足测试点 15 ∼ 17 的限制。

【样例 5】

见选手文件中的 ***control/control5.in*** 与 ***control/control5.ans***。
此样例满足测试点 18 ∼ 21 的限制。

【数据范围】

对于 100% 的数据，$2 \le n \le 3\,000$，$1 \le k \le \min(n, 1\,000)$，$1 \le q \le 5 \times 10^5$。

<table>
<tr><th>测试点编号</th><th>$n \le$</th><th>$k \le$</th><th>$q \le$</th><th>特殊性质</th></tr>
<tr><td>$1 \sim 3$</td><td>80</td><td>50</td><td>2 000</td><td>无</td></tr>
<tr><td>$4 \sim 6$</td><td rowspan="3">400</td><td>300</td><td>1</td><td>k 是偶数</td></tr>
<tr><td>$7 \sim 10$</td><td>2</td><td rowspan="2">10 000</td><td rowspan="3">无</td></tr>
<tr><td>$11 \sim 14$</td><td>300</td></tr>
<tr><td>$15 \sim 17$</td><td rowspan="3">3 000</td><td>10</td><td rowspan="3">5×10^5</td></tr>
<tr><td>$18 \sim 21$</td><td rowspan="2">1 000</td><td>k 是偶数</td></tr>
<tr><td>$20 \sim 25$</td><td>无</td></tr>
</table>