

2024 年全国青少年信息学联合省选模拟赛

重庆北站搬

第一试

时间: 2024 年 1 月 5 日 08:00 ~ 13:00

题目名称	剪彩灯	叉集合	玩图
题目类型	传统型	传统型	传统型
目录	lamp	set	graph
可执行文件名	lamp	set	graph
输入文件名	lamp.in	set.in	graph.in
输出文件名	lamp.out	set.out	graph.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	1536 MB
测试点数目	20	4	5
测试点是否等分	是	否	否

提交源程序文件名

对于 C++ 语言	lamp.cpp	set.cpp	graph.cpp
-----------	----------	---------	-----------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11
-----------	--------------------

注意事项与提醒 (请选手务必仔细阅读)

1. 选手提交的源程序必须存放在已建立好的, 且带有**样例文件**和**下发文件**的文件夹中, 文件夹名称与对应试题英文名一致。
2. 文件名 (包括程序名和输入输出文件名) 必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int, 值必须为 0。
4. **对于因未遵守以上规则对成绩造成的影响, 相关申诉不予受理。**
5. 若无特殊说明, 结果比较方式为**忽略行末空格、文末回车后的全文比较**。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大, 但你使用的栈空间大小不应超过题目限制。
8. 每道题目所提交的**代码文件大小限制为 100KB**。
9. 若无特殊说明, 输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
10. 输入文件中可能存在行末空格, 请选手使用更完善的读入方式 (例如 `scanf` 函数) 避免出错。
11. 直接复制 PDF 题面中的多行样例, 数据将带有行号, 建议选手直接使用对应目录下的样例文件进行测试。

12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。
13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外不允许在程序中手动开启其他编译选项，一经发现，本题成绩以 0 分处理。

剪彩灯 (lamp)

【题目描述】

有一串长度为 n 的彩灯，其中第 i 盏彩灯的颜色为 a_i 。小 A 将这串彩灯剪成若干段并装饰在房间里，每一段彩灯的美丽度为这段彩灯的颜色形成的集合的 mex，即第一个未出现在集合内的非负整数，例如 $\text{mex}\{1, 2, 4\} = 0$, $\text{mex}\{0, 1, 2, 4\} = 3$ 。

由于彩灯之间的奇特相互作用，整个房间的美丽度为每段彩灯的美丽度的积。小 A 想知道所有剪彩灯的方案的美丽度和是多少。若两个方案中存在一对相邻的彩灯间，一个方案剪断，一个方案未剪断则视为不同的方案。答案对 998244353 取模。

【输入格式】

从文件 *lamp.in* 中读入数据。

第一行一个正整数 n 。

第二行 n 个非负整数，第 i 个数 a_i 代表第 i 盏彩灯的颜色。

【输出格式】

输出到文件 *lamp.out* 中。

一行一个数，表示答案对 998244353 取模后的值。

【样例 1 输入】

```
1 4
2 0 1 0 2
```

【样例 1 输出】

```
1 8
```

【样例 2 输入】

```
1 10
2 1 0 2 0 3 1 0 2 3 4
```

【样例 2 输出】

```
1 133
```

【样例 3】

见下发文件。

【数据范围】

对于 100% 的数据满足: $1 \leq n \leq 10^6, 0 \leq a_i \leq n$ 。

对于前 30% 的数据, $n \leq 5000$ 。

对于前 60% 的数据, $a_i \leq 5000$ 。

叉集合 (set)

【题目描述】

小 B 有一个集合 S ，初始是空集。现在有 n 次操作，每次操作为以下两种之一：

1. 向集合 S 中插入一个数 w ，保证此时集合 S 中没有 w 。
2. 选择区间 $[0, w]$ 中的一个数 x 和集合 S 中互不相同的两个数 i, j ，你需要最小化 $(x + i) \oplus (x + j)$ 并输出这个最小值。其中 \oplus 表示二进制下的异或，保证此时集合 S 中有至少两个数。

【输入格式】

从文件 `set.in` 中读入数据。

第一行两个正整数 V, n ，其中 V 在数据范围中有介绍。

接下来 n 行，每行两个整数 op, w ，其中 op 表示第几个操作。

【输出格式】

输出到文件 `set.out` 中。

对于每一个 2 操作输出一行一个整数表示答案。

【样例 1 输入】

```
1 7 5
2 1 85
3 1 69
4 1 24
5 1 82
6 2 71
```

【样例 1 输出】

```
1 3
```

【样例 1 解释】

$x = 2, i = 82, j = 85$ 时异或值最小。

【样例 2 输入】

```
1 7 10
```

```
2 1 2
3 1 50
4 2 121
5 2 58
6 1 90
7 2 86
8 1 117
9 2 2
10 1 67
11 2 75
```

【样例 2 输出】

```
1 48
2 48
3 40
4 43
5 17
```

【样例 3】

见下发文件。

【数据范围】

对于 100% 的数据，有 $3 \leq n \leq 10^5, 1 \leq op \leq 2, 0 \leq w < 2^V$ 。

测试点编号	分值	$V =$	$n \leq$	特殊性质
1	15	7	10^2	无
2	20	20	10^5	只有最后一次操作 $op=2$
3	30	20	10^5	无
4	35	40	10^5	无

玩图 (graph)

【题目描述】

小 C 有一张无向图，最开始图仅有一个 0 号节点。现在有 n 次操作，每次操作为以下 5 种之一（不妨假设每次操作前这张图的节点编号区间为 $[l, r]$ ）：

1. 删去 l 号节点，并删去 l 号节点连接的所有边。
2. 删去 r 号节点，并删去 r 号节点连接的所有边。
3. 增加 $l - 1$ 号节点，并连接 $\min(k - 1, r - l + 1)$ 条边，第 i 条边连接 $(l - 1, l - 1 + i)$ ，边有边权。
4. 增加 $r + 1$ 号节点，并连接 $\min(k - 1, r - l + 1)$ 条边，第 i 条边连接 $(r + 1, r + 1 - i)$ ，边有边权。
5. 对当前图询问最小生成树的边权和。

输入保证任意时刻 $l \leq r$ 。

【输入格式】

从文件 *graph.in* 中读入数据。

第一行三个正整数 $seed, k, n$ ，其中 $seed$ 表示随机数生成器的种子。

接下来 n 行，每行一个正整数 op ，表示第几个操作。当 $op = 3/4$ 时，为了减少输入，你需要调用 $\min(k - 1, r - l + 1)$ 次随机数生成器来获得边权，第 i 次调用表示第 i 条边的边权。

随机数生成器：

```
1 namespace qwq{
2     std::mt19937 eng;
3     void init(int Seed){eng.seed(Seed);}
4     int readW(){
5         return uniform_int_distribution<int>(0,1000000000)(eng);
6     }
7 }
```

当你输入了 $seed$ 后需要调用 `qwq::init(seed)` 来初始化，获得边权时调用 `qwq::readW()`。

【输出格式】

输出到文件 *graph.out* 中。

对于每一个 5 操作输出一行一个整数表示答案。

【样例 1 输入】

```
1 20220220 4 4
2 4
3 4
4 4
5 5
```

【样例 1 输出】

```
1 1139655038
```

【样例 1 解释】

询问时有边：

```
1 (0,1,780392573)
2 (1,2,852196855)
3 (0,2,494487013)
4 (2,3,57484417)
5 (1,3,895195425)
6 (0,3,301778048)
```

【样例 2 输入】

```
1 20220220 4 12
2 4
3 3
4 4
5 5
6 2
7 2
8 3
9 3
10 2
11 5
12 1
13 5
```

【样例 2 输出】

- 1

1139655038
- 2

1079079398
- 3

416038443

【样例 3】
见下发文件。

【数据范围】
对于 100% 的数据，有 $2 \leq k \leq 10, 1 \leq n \leq 5 \times 10^5, 1 \leq op \leq 5, 1 \leq seed \leq 10^9$ 。

测试点编号	分值	$n \leq$	特殊性质
1	15	10^3	无
2	20	10^5	没有操作 1, 2
3	20	10^5	没有操作 1, 3
4	20	10^5	前 k 次操作均为 4 操作，且之后的任意时刻 $l \leq 0, k \leq r$
5	25	5×10^5	无