

广东省队集训 2025 NOI 模拟赛

广东省集

模拟赛

题目名称	火力全开	异或症测试 3	树上邻邻域数点
题目类型	传统型	传统型	传统型
目录	fire	basis	tree
可执行文件名	fire	basis	tree
输入文件名	fire.in	basis.in	tree.in
输出文件名	fire.out	basis.out	tree.out
每个测试点时限	4.0 秒	1.0 秒	2.0 秒
内存限制	1024 MiB	512 MiB	1024 MiB
测试点数目	0	0	0
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	fire.cpp	basis.cpp	tree.cpp
-----------	----------	-----------	----------

编译选项

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

注意事项（请仔细阅读）

测试点数目与是否等分以题目为准。

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。

火力全开 (fire)

【题目描述】

有 n 个敌人，你有两种攻击他们的方式：

1. 花费 1 的代价，选择一个敌人，对其进行一次普通攻击。
2. 使用一颗炮弹，对所有敌人造成一次爆炸。

对于第二种攻击方式，有 m 颗炮弹可供使用，使用第 i 颗炮弹需要花费 c_i 的代价，并造成一次威力为 d_i 的爆炸。每颗炮弹只能使用一次。

对于第 i 个敌人，如果其被使用第一种攻击方式攻击了 a_i 次，或者受到了 k 次威力不小于 b_i 的爆炸，就会死亡。

有 q 次修改，每次输入 op, x, y, z ，如果 $op = 1$ ，表示将 a_x 修改为 y ， b_x 修改为 z 。否则 $op = 2$ ，表示将 c_x 修改为 y ， d_x 修改为 z 。每次修改后，求出至少需要花费多少代价，才能使所有敌人死亡。修改之间不独立，也即修改的结果会继承。

【输入格式】

第一行四个正整数 n, m, q, k ，分别表示敌人个数，炮弹颗数，修改次数，敌人的抗爆属性。

接下来 n 行，每行两个整数 a_i, b_i ，表示敌人的属性。

接下来 m 行，每行两个整数 c_i, d_i ，表示炮弹的属性。

接下来 q 行，每行四个整数 op, x, y, z ，表示一次修改。

【输出格式】

输出 q 行，每行一个整数，表示每次修改后的答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 10
2 16
3 14
```

【样例 1 解释】

第一次修改后，直接选择第一颗炮弹，可以将所有敌人炸死，代价为 10。

第二次修改后，最优方案是用普通攻击击败所有敌人，代价为 16。

第三次修改后，使用第二颗炮弹炸掉第一个敌人，再用普通攻击击败第二个敌人，代价为 14。

【样例 2~3】

见下发文件，分别满足子任务 1,5 的限制条件。

【子任务】

对于所有数据， $1 \leq n, m, q \leq 2.5 \times 10^5$ ， $1 \leq k \leq 10^4$ ， $qk \leq 5 \times 10^5$ ， $1 \leq op \leq 2$ ， $1 \leq a_i, b_i, c_i, d_i, y, z \leq 10^9$ ，如果 $op = 1$ ，则 $1 \leq x \leq n$ ，否则 $1 \leq x \leq m$ 。

子任务编号	$n, m, q \leq$	特殊性质	分数
1	200	无	10
2	5000	无	20
3	10^5	$op = 1$	20
4	10^5	$k \leq 10$	20
5	2.5×10^5	无	30

异或症测试 3 (basis)

【题目描述】

给你一个集合 B 和一个整数 X ，你要求出有多少个非空集合 $S \subseteq \{1, 2, 3, \dots, X\}$ ，使得 B 是 S 的线性基。答案对 998244353 取模。

称一个集合 B 是另一个集合 S 的线性基，当且仅当 S 中任意元素都能被表示为 B 中若干元素的异或和，且 B 是所有这样的集合中大小最小的（可能有多个最小的）。

【输入格式】

第一行两个整数 n, m ， n 为集合 B 的大小， B 中所有元素以及 X 均不超过 $2^m - 1$ 。不保证 B 中没有重复元素。

接下来 n 行，每行一个长为 m 的 01 字符串，第 i 行为 B_i 的二进制表示，行末为最低位。

最后一行一个长为 m 的 01 字符串，为 X 的二进制表示。

【输出格式】

输出一行一个整数，表示答案对 998244353 取模的结果。

【样例 1 输入】

```
1 2 2
2 01
3 10
4 11
```

【样例 1 输出】

```
1 4
```

【样例 1 解释】

$B_1 = 1$ ， $B_2 = 2$ ， $X = 3$ ，集合 $\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$ 是合法的。

【样例 2 输入】

```
1 3 3
2 001
3 010
4 101
5 110
```

【样例 2 输出】

```
1 38
```

【样例 3~5】

见下发文件，分别满足子任务 2, 4, 7 的限制。

【子任务】

对于所有数据， $1 \leq n, m \leq 2000$ 。

子任务编号	$n, m \leq$	特殊性质	分数
1	4	无	10
2	10	无	10
3	30	无	10
4	100	A	10
5	500	无	20
6	2000	A	20
7	2000	无	20

特殊性质 A：保证 $n = m$ ， $X = 2^m - 1$

树上邻邻域数点 (tree)

【题目描述】

这是一道交互题。保证在询问合法的情况下，交互库占用运行时间不超过 500ms，空间不超过 128MB。因此你的程序运行时间不应超过 1500ms，空间不超过 896MB。

给你一棵 N 个点的树，节点编号为 $0 \sim N-1$ ，点 i 有一个未知的整数点权 a_i ，保证 $0 \leq a_i < 32$ 。每次你可以给出整数 x, d, v ，向交互库询问：在树上距离 x 恰为 d 的点中，有多少个点 y 满足 $a_y \leq v$ 。你需要在不超过 M 次询问内求出每个点的点权。并且询问的 d 不小于给定的限制 L 。保证不存在一个点的度数为 $N-1$ 。定义树上两点的距离为两个点最短路径经过的边数。

【实现细节】

选手不需要，也不应该实现 main 函数。选手应确保提交的程序包含头文件 `tree.h`，可在程序开头加入以下代码实现：

```
1 #include "tree.h"
```

选手需要实现以下函数：

```
1 std::vector<int> tree(int N, std::vector<std::pair<int,int> >
    E, int M, int L);
```

- N 表示树的节点个数。
- E 表示树的边集，大小为 $N-1$ ，其中的元素 (u, v) 表示存在一条连接 u 和 v 的边；
- M 表示询问次数限制。
- L 表示询问的 d 的限制。
- 该函数需要返回长为 N 的数组 ret ，编号 $0 \sim N-1$ ，其中 $ret_i = a_i$ 。
- 对于每个测试点，该函数会被交互库调用恰好 1 次。

选手可以通过调用以下函数向交互库发送一次询问：

```
1 int ask(int x, int d, int v);
```

- 你需要确保 $0 \leq x < N$ ， $L \leq d \leq N$ ， $0 \leq v < 32$ 。
- 该函数会返回在树上距离 x 恰为 d 的点 y 中，满足 $a_y \leq v$ 的点的个数。

【测试程序方式】

下发文件中的 `template_tree.cpp` 是一份示例代码，`grader.cpp` 是提供的交互库参考实现，最终测试时所用的交互库实现与该参考实现有所不同，因此选手的解法不应该依赖交互库的实现。

选手可以在本题目录下使用如下命令编译得到可执行程序:

```
1 g++ grader.cpp tree.cpp -o tree -O2 -std=c++14 -static
```

对于编译得到的可执行程序:

- 可执行文件将从标准输入读入以下格式的数据:
 - 输入的第一行包含三个非负整数 N, M, L , 表示树的节点个数和询问限制。
 - 接下来一行输入一个非负整数 R_1 , 表示树的生成方式, 如果 $R_1 = 0$, 则接下来 $N - 1$ 行, 每行读入两个数 u, v , 表示树上的边。否则将会以 R_1 为随机种子, 对 $1 \leq i < N$ 随机生成 $0 \leq fa_i < i$, 树上的每条边为 (fa_i, i) 。
 - 接下来一行输入一个非负整数 R_2 , 表示点权的生成方式, 如果 $R_2 = 0$, 则接下来一行读入 N 个数, 第 i 个数为 a_i , 表示 i 的点权。否则将会以 R_2 为随机种子, 对 $0 \leq i < N$ 随机生成 $0 \leq a_i < 32$ 。

【子任务】

本题共有 3 个子任务。所有数据均满足 $N = 50000$ 。

子任务编号	$M =$	$L =$	分数
1	2.5×10^5	0	10
2	10^6	1	30
3	10^6	2	60

其中, 对于子任务 2 和子任务 3, 有更特别的评分方式, 假设你实际询问次数为 Q , 那么你可以获得的分数占该子任务满分的百分比为:

Q	百分比
$Q \leq 2.5 \times 10^5$	100
$2.5 \times 10^5 < Q \leq 3 \times 10^5$	$80 + \lfloor (3 \times 10^5 - Q) / 2500 \rfloor$
$3 \times 10^5 < Q \leq 5 \times 10^5$	$40 + \lfloor (5 \times 10^5 - Q) / 5000 \rfloor$
$5 \times 10^5 < Q \leq 10^6$	$\lfloor (10^6 - Q) / 12500 \rfloor$