

正睿青少年信息学奥林匹克竞赛

2025 暑假 AB 班联考

Day3

时间：2025 年 8 月 8 日 8:00 ~ 12:00

题目名称	序列	图	测试
题目类型	传统题	传统题	传统题
目录	seq	graph	test
可执行文件名	seq	graph	test
输入文件名	seq.in	graph.in	test.in
输出文件名	seq.out	graph.out	test.out
每个测试点时限	1.0 秒	2.0 秒	5.0 秒
内存限制	1024 MB	1024 MB	1024 MB
测试点数目	8	10	6
测试点是否等分	否	否	否

提交源程序文件名

对于 C++ 语言	seq.cpp	graph.cpp	test.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。赛后正式测试时将以选手留在题目目录下的源代码为准。
2. `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. 因违反以上两点而出现的错误或问题，申诉时一律不予受理。
4. 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）。
5. 选手提交的程序源文件必须不大于 100KB。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 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。
8. 选手可在比赛主页绑定工作目录后点击【保存预提交】将工作目录下所有代码一次性提交到评测系统。

序列 (seq)

【题目描述】

给定两个整数 m, k ，请你求出满足以下所有条件的最长正整数序列 $a = (a_1, a_2, \dots, a_n)$ ，并输出最大的可能长度 n 。

- 1. 对于所有 i ，满足 $1 \leq a_i \leq m$
- 2. 对于所有 $i \neq j$ ，满足 $a_i \neq a_j$ （即所有元素互不相同）
- 3. 从序列中任意选择 k 个或更多个不同元素，它们的最大公约数为 1

【输入格式】

输入一行包含两个整数 m 和 k 。

【输出格式】

输出满足条件的最大长度 n 。

【样例 1 输入】

1

11 5

【样例 1 输出】

1

10

【样例 1 解释】

$a = [1, 2, 3, 4, 5, 7, 8, 9, 10, 11]$ 是一种方案。

【样例 2 输入】

见下发文件中的 seq2.in

【样例 2 输出】

见下发文件中的 seq2.ans

【样例 3 输入】

见下发文件中的 seq3.in

【样例 3 输出】

见下发文件中的 seq3.ans

【样例 4 输入】

见下发文件中的 seq4.in

【样例 4 输出】

见下发文件中的 seq4.ans

【样例 5 输入】

见下发文件中的 seq5.in

【样例 5 输出】

见下发文件中的 seq5.ans

【数据范围与提示】

$1 \leq m \leq 2 \times 10^5, 2 \leq k \leq 5。$

subtaskid	$k =$	$m \leq$	分值
1	2	100	5
2	2	2×10^5	5
3	3	100	10
4	3	2×10^5	10
5	4	100	15
6	4	2×10^5	15
7	5	100	20
8	5	2×10^5	20

图 (graph)

【题目描述】

有一个 n 个点 m 条边的简单联通无向带权图，其中 m_0 条边已经被给定，保证每条边的边权都在 $[0, 2^w)$ 之间。

定义一个 n 个点 m 条边的简单联通无向带权图是好的，当且仅当满足以下条件：

1. 所有边的边权都在 $[0, 2^w)$ 之间。
2. 包含给定的边。
3. 忽略掉边权后仍然不存在重边。
4. 联通。
5. 对于任意两个点 x, y 和 $z \in [0, 2^w)$ ，都存在一条路径（不一定简单）从 x 到 y 且边的异或和为 z （重复经过一条边计算多次）。

你要统计有多少个有标号的图是好的，答案对 998244353 取模。

【输入格式】

第一行四个整数，表示 n, m, m_0, w 。

接下来 m_0 行，每行三个整数 $u, v, weight$ 表示一条边。

【输出格式】

一行，表示好的图数量，对 998244353 取模。

【样例 1 输入】

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

【样例 1 输出】

```
1 2
```

【样例 1 解释】

三条边必须出现，并且另外两条边的边权不可以相同。

【样例 2 输入】

```
1 5 7 3 2
2 1 2 3
3 2 3 3
4 1 3 0
```

【样例 2 输出】

```
1 3360
```

【样例 3 输入】

见下发文件中的 graph3.in

【样例 3 输出】

见下发文件中的 graph3.ans

【样例 4 输入】

见下发文件中的 graph4.in

【样例 4 输出】

见下发文件中的 graph4.ans

【数据范围与提示】

对于所有数据, 满足 $1 \leq n \leq 50, n-1 \leq m \leq \frac{n(n-1)}{2}, 0 \leq m_0 \leq m, 0 \leq w \leq 60, 1 \leq u, v \leq n, 0 \leq weight < 2^w$

subtaskid	$n \leq$	$w \leq$	分值
1	10	0	10
2	10	60	10
3	20	0	10
4	20	60	10
5	30	0	10
6	30	60	10
7	40	0	10
8	40	60	10
9	50	0	10
10	50	60	10

测试 (test)

【题目描述】

你有 n 堆石子，排成一排。第 i 堆有 a_i 个石子。在所有石子中，恰好有一颗是特殊石子，你的目标是找出它的位置。

每次测试，你可以从某一堆中选择若干颗石子进行测试。测试结果会告诉你：所选石子中是否包含那颗特殊石子。注意，每次测试只能从一堆中选石子。

你希望在最坏情况下，用尽量少的测试次数找出这颗特殊石子。

现在有 q 个询问，每个询问给定一段区间 $[l, r]$ 。对于每个询问，要求你输出：若已知特殊石子一定在第 l 到第 r 堆之间，最少需要多少次测试，才能确定它的位置。

由于 a_i 的值可能很大，输入中用长度为 m 的二进制字符串来表示每个 a_i 。字符串从高位到低位排列。

【输入格式】

第一行包含三个正整数 n 、 m 和 q ，分别表示石子堆的数量、二进制表示的最大长度，以及询问的数量。

接下来 n 行，每行一个长度为 m 的二进制字符串，表示每一堆的石子数量 a_i （高位在前）。

接下来 q 行，每行两个整数 l 和 r ，表示一个查询区间。

【输出格式】

对于每个询问，输出在最坏情况下确定特殊石子所需的最小测试次数。

【样例 1 输入】

```
1 5 3 4
2 001
3 011
4 010
5 011
6 001
7 3 3
8 4 5
9 1 5
10 1 3
```

【样例 1 输出】

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

【样例 1 解释】

$a = [1, 3, 2, 3, 1]$ 。

对于第一组询问，只需要把同一堆的两个分开检验一次即可。

对于第二组询问，最少需要两次询问，策略如下：

1. 第一次操作从3个石子的一堆中选取两个。
2. 无论 1.中的回答是什么，我们都可以把答案锁定在2个石子之一，再花费一次询问即可。

【样例 2 输入】

见下发文件中的 `test2.in`

【样例 2 输出】

见下发文件中的 `test2.ans`

【样例 3 输入】

见下发文件中的 `test3.in`

【样例 3 输出】

见下发文件中的 `test3.ans`

【样例 4 输入】

见下发文件中的 `test4.in`

【样例 4 输出】

见下发文件中的 `test4.ans`

【样例 5 输入】

见下发文件中的 `test5.in`

【样例 5 输出】

见下发文件中的 test5.ans

【数据范围与提示】

对于所有数据 $1 \leq n, q \leq 5 \times 10^4, 1 \leq m \leq 1000, 1 \leq a_i < 2^m$ 。

subtaskid	$n, q \leq$	$m \leq$	分值
1	10	10	10
2	100	20	20
3	100	100	10
4	2000	200	20
5	30000	1000	20
6	50000	1000	20