

NOIP 模拟赛

题目名称	困难的题目	优秀的赛制	公平的竞争	完美的答卷
题目类型	传统型	传统型	传统型	传统型
目录	hard	good	fair	perfect
可执行文件名	hard	good	fair	perfect
输入文件名	hard.in	good.in	fair.in	perfect.in
输出文件名	hard.out	good.out	fair.out	perfect.out
每个测试点时限	0.5 秒	0.5 秒	0.1 秒	0.5 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	8	10	9	6
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	hard.cpp	good.cpp	fair.cpp	perfect.cpp
-----------	----------	----------	----------	-------------

编译选项

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

【注意事项】

1. 选手提交的源程序请**直接放在个人目录下**，无需建立子文件夹。
2. 选手提交的源程序**大小不应超过 100KB**。
3. 文件名（程序名和输入输出文件名）必须使用英文小写。
4. main() 函数的返回类型必须是 int，程序正常结束时的返回值必须是 0。
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. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. **请不要复制题面中的任何文本**，以免出现问题。
8. **本场比赛开启捆绑测试**，选手提交的代码必须通过一个子任务内的所有测试数据才能获得该子任务的分数，且子任务之间有合理的依赖关系。

困难的题目 (hard)

【题目背景】

F 不喜欢模拟赛，压力的来源正是困难的题目。

【题目描述】

教练准备了 n 场模拟赛，每场模拟赛会考察 m 个知识点中的一部分。

教练希望在讲解知识点后的一段时间内考察它，也就是说，他会在第 l_i 到第 r_i 场模拟赛考察第 i 个知识点。

F 对每个知识点都有自己的难度评分，他认为第 i 个知识点的难度是 v_i ，一场模拟赛的难度为它考察的所有知识点的难度评分之和，自然地，**如果一场模拟赛没有考察任何知识点，那么它的难度是 0。**

F 已经学过了一些知识点，而对于没学过的知识点，他可以将难度评估为在 $[0, w]$ 之内的任意整数。对于每场模拟赛，他想知道是否存在一种评估难度的方式，使得这场模拟赛是 n 场模拟赛中难度最高的。

【输入格式】

从文件 *hard.in* 中读入数据。

第一行三个正整数 n, m, w ，分别表示模拟赛、知识点的数量和难度的值域。

接下来 m 行描述 m 个知识点，第 i 行三个整数 l_i, r_i, v_i 表示第 i 个知识点的出现时间和难度评分。其中 $v_i = -1$ 表示 F 没学过这个知识点， $v_i \neq -1$ 表示 F 学过这个知识点并认为它的难度为 v_i 。

【输出格式】

输出到文件 *hard.out* 中。

一个长度为 n 的 01 串，第 i 位为 1 当且仅当这场模拟赛可能是 n 场模拟赛中难度最高的。

【样例 1 输入】

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

【样例 1 输出】

1 11

【样例 1 解释】

令 $v_2 = 1, v_3 = 2$, 此时两场模拟赛的难度均为 4, 均是难度最高的。

【样例 2】

见选手目录下的 *hard/hard2.in* 与 *hard/hard2.ans*。
该样例满足子任务 4 的限制。

【样例 3】

见选手目录下的 *hard/hard3.in* 与 *hard/hard3.ans*。
该样例满足子任务 5 的限制。

【样例 4】

见选手目录下的 *hard/hard4.in* 与 *hard/hard4.ans*。
该样例满足子任务 6 的限制。

【样例 5】

见选手目录下的 *hard/hard5.in* 与 *hard/hard5.ans*。
该样例满足子任务 8 的限制。

【数据范围】

对于 100% 的数据, $1 \leq n, m \leq 3 \times 10^5, 1 \leq w \leq 10^6, -1 \leq v_i \leq w$ 。

子任务	$n \leq$	$m \leq$	特殊性质	得分
1	10	5	$w \leq 1$	10
2	500	500		15
3	5000	5000	$w \leq 1$	10
4				20
5	3×10^5	3×10^5	$n = m, l_i = r_i = i$	10
6			$v_i \neq -1$	5
7			$w \leq 1$	10
8				20

优秀的赛制 (good)

【题目背景】

SelfEval 真是 NOI2023 最伟大的发明。

只是 F 的实力还配不上这优秀的赛制。

【题目描述】

F 想要 AC 一道有 n 个子任务的题，这些子任务的包含关系恰好构成了一颗有根树，其中对于所有 $u \neq 1$ 均有一个 p_u 满足 $1 \leq p_u < u$ 表示 u 在树上的父亲。

称子任务 u 包含子任务 v 当且仅当 u 是 v 的祖先，特别地，子任务 u 包含它自己。

F 最多有 k 次使用 SelfEval 的机会，每次他只能测试一个子任务是否正确。

F 想用完 k 次提交机会，且不希望有无用提交，故称一个提交序列 a 是合理的，当且仅当对于所有 $1 \leq j < i \leq k$ ，子任务 a_j 不包含子任务 a_i 。

几次提交过后 F 遇到了一个问题，那就是他不能保证每次提交都能通过该子任务。第 i 次提交有 50% 的概率通过子任务 a_i ，但若存在 $1 \leq j < i$ 满足 $p_{a_j} = a_i$ 且 F 通过了 a_j ，那么他有 100% 的概率通过 a_i 。

F 想知道对于所有 $k = 1, 2, \dots, K$ ，所有长度为 k 的合理提交序列 a 能通过这道题（子任务 1）的概率之和，答案对 $10^9 + 7$ 取模。

【输入格式】

从文件 `good.in` 中读入数据。

第一行两个正整数 n, K ，分别表示子任务数量和询问范围。

第二行 $n - 1$ 个正整数，第 i 个正整数 p_{i+1} 表示 $i + 1$ 在树上的父亲。

【输出格式】

输出到文件 `good.out` 中。

共 K 行，第 i 行表示 $k = i$ 的答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 500000004
2 750000007
3 750000008
```

【样例 1 解释】

对于 $k = 1$, 有 4 个合法序列 (1), (2), (3), (4), 其中只有 (1) 提交了子任务 1, 且有 50% 的概率通过, 故答案为 $\frac{1}{2}$ 。

对于 $k = 2$, 有 7 个合法序列 (2, 1), (3, 1), (3, 2), (3, 4), (4, 1), (4, 2), (4, 3), 其中只有 (2, 1), (3, 1), (4, 1) 提交了子任务 1, 且分别有 75%, 50%, 50% 的概率通过, 故答案为 $\frac{7}{4}$ 。

对于 $k = 3$, 有 6 个合法序列, 答案为 $\frac{11}{4}$ 。

【样例 2】

见选手目录下的 *good/good2.in* 与 *good/good2.ans*。
该样例满足子任务 2 的限制。

【样例 3】

见选手目录下的 *good/good3.in* 与 *good/good3.ans*。
该样例满足子任务 6 的限制。

【样例 4】

见选手目录下的 *good/good4.in* 与 *good/good4.ans*。
该样例满足子任务 8 的限制。

【样例 5】

见选手目录下的 *good/good5.in* 与 *good/good5.ans*。
该样例满足子任务 9 的限制。

【样例 6】

见选手目录下的 *good/good6.in* 与 *good/good6.ans*。
该样例满足子任务 10 的限制。

【数据范围】

对于 100% 的数据, $1 \leq K \leq n \leq 5000, 1 \leq p_i < i$ 。

子任务	$n \leq$	$K \leq$	特殊性质	得分
1	8	n		16
2	50			12
3	200	10		8
4		n	$p_i = i - 1$	8
5			$p_i = 1$	8
6				8
7	5000	10		8
8		n	$p_i = i - 1$	8
9			$p_i = 1$	8
10				16

公平的竞争 (fair)

【题目背景】

所以要“费厄”，最好是首先看清对手，倘是些不配承受“费厄”的，大可以老实不客气；待到它也“费厄”了，然后再与它讲“费厄”不迟。（节选自鲁迅《论“费厄泼赖”应该缓行》）

作为学科竞赛的不可忽视的特点，公平性的受关注度极高。

所以作为选择题的选项来说，它们也应该享有权利进行公平的竞争。

【题目描述】

F 在做选择题，这道题有 n 个选项，而他手里只有一个 k 面的骰子（如果 $k = 2$ 则为硬币）。

为了落实公平竞争的理念，他希望他的策略选择到每个选项的概率相等。

求他期望投掷次数的最小值，答案对质数 M 取模。

【输入格式】

从文件 `fair.in` 中读入数据。

一行三个正整数 n, k, M ，分别表示选项数、骰子面数和模数。

【输出格式】

输出到文件 `fair.out` 中。

一个整数表示模意义下的答案。

【样例 1 输入】

```
1 3 2 998244353
```

【样例 1 输出】

```
1 665496238
```

【样例 1 解释】

投两次硬币，得到 4 种等概的结果，前三种结果分别表示选择三种选项，最后一种结果重复执行这个过程。

设答案为 x ，有 $x = 2 + \frac{x}{4}$ ，解得 $x = \frac{8}{3}$ 。

【样例 2 输入】

1 10 2 998244353

【样例 2 输出】

1 798595487

【样例 2 解释】

答案为 $\frac{23}{5}$ 。

【样例 3 输入】

1 2991671 1575767 100000123

【样例 3 输出】

1 69532098

【样例 4 输入】

1 2646870 5 998244353

【样例 4 输出】

1 593057467

【样例 5 输入】

1 725 26 998244353

【样例 5 输出】

1 915983925

【样例 6 输入】

1 744489 9 100000073

【样例 6 输出】

1 89453674

【数据范围】

对于 100% 的数据， $2 \leq k \leq n \leq 3 \times 10^6$, $10^8 \leq M \leq 10^9$ ，保证 M 为质数。

子任务	$k \leq$	$n \leq$	特殊性质	得分
1	2	10	AC	5
2		3×10^6	AB	15
3			AC	15
4	AB		5	
5	AC		5	
6	5000		5000	
7	3×10^6	3×10^6	B	10
8			C	10
9				20

特殊性质 A: 保证 k 为质数。

特殊性质 B: 保证 n, k 互质。

特殊性质 C: 保证 $n = k^p q$ ，其中 p 为非负整数且 q, k 互质。

【提示】

可以证明在 M 是质数时答案在模意义下存在。

完美的答卷 (perfect)

【题目背景】

F 还是退役了，成为了超级银牌王。

也许多年以后他会忘掉排名，忘掉题目，忘掉 OI，甚至忘掉 A+B 怎么写。

但他永不会忘记那些热血沸腾、激动人心的瞬间。

这份绝对的炽热，为他的 OI 生涯交出了完美的答卷。

【题目描述】

F 的生涯一共可以分成 n 个阶段，每个阶段有一个实力值 a_i ，保证实力值之间互不相同。

F 用一种奇怪的方式给他的每个生涯区间打了分。

对于 $1 \leq l \leq r \leq n$ ，生涯区间 $[l, r]$ 的分数 $f(l, r) = \max_{i=l}^r \{a_i\} \oplus \min_{i=l}^r \{a_i\}$ ，其中 \oplus 表示按位异或。

F 想知道所有生涯区间的分数的最大值，也就是 $\max_{1 \leq l \leq r \leq n} \{f(l, r)\}$ 。

这是他生涯中最后的愿望了。

【输入格式】

从文件 `perfect.in` 中读入数据。

第一行一个正整数 n ，表示阶段数量。

第二行 n 个非负整数，第 i 个整数 a_i 表示第 i 个阶段的实力值。

【输出格式】

输出到文件 `perfect.out` 中。

一个整数表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 7
```

【样例 2】

见选手目录下的 *perfect/perfect2.in* 与 *perfect/perfect2.ans*。
该样例满足子任务 2 的限制。

【样例 3】

见选手目录下的 *perfect/perfect3.in* 与 *perfect/perfect3.ans*。
该样例满足子任务 3 的限制。

【样例 4】

见选手目录下的 *perfect/perfect4.in* 与 *perfect/perfect4.ans*。
该样例满足子任务 5 的限制。

【数据范围】

对于 100% 的数据, $1 \leq n \leq 3 \times 10^5, 0 \leq a_i \leq 10^6$, 保证 a_i 互不相同。

子任务	$n \leq$	特殊性质	得分
1	500		12
2	5000		16
3	3×10^5	A	8
4		B	16
5		C	20
6			28

特殊性质 A: 保证对于所有 $1 \leq i < n$, 有 $a_i < a_{i+1}$ 。

特殊性质 B: 保证对于所有 $1 \leq i < n$, 有 $a_i \& a_{i+1} = 0$, 其中 $\&$ 表示按位与。

特殊性质 C: 保证数据随机。