

2020 年高中信息学多校联合训练

MSTI2020 模拟赛

Day 2

时间：2020 年 11 月 4 日 8:00 ~ 12:00

题目名称	数据恢复	下落的小球	消失的运算符	古老的序列问题
题目类型	传统型	传统型	传统型	传统型
目录	data	ball	operator	sequence
可执行文件名	data	ball	operator	sequence
输入文件名	data.in	ball.in	operator.in	sequence.in
输出文件名	data.out	ball.out	operator.out	sequence.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	1024 MB	512 MB
子任务数目	6	8	5	10
测试点是否等分	否	否	是	是

提交源程序文件名

对于 C++ 语言	data.cpp	ball.cpp	operator.cpp	sequence.cpp
-----------	----------	----------	--------------	--------------

编译选项

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

注意事项：

1. 测评时栈空间与内存限制相同。
2. 时间限制保证在标程的两倍以上，具体时限可随实际测评环境调整。
3. 若无特殊说明，输入文件的同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
4. 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。
5. **题目按照题目名字长度排序，与难度无关。请自行选择写题顺序。**
6. 虽然还没到 2022 年，但是测评环境无法支持 Pascal 语言，希望谅解。
7. 题目比较简单，请独立完成。

数据恢复(data)

【题目背景】

宫水三叶被要求出一场测试，于是三叶非常认真的出了一套题。

然而在临近测试时，一场意外使得这场测试所有的资料全部被清空了。其中包括题面，题解，标程和数据。

无奈之下，三叶找到了小 H，希望她能恢复这些被清空的数据。

【题目描述】

擅长电脑的小 H 明白，这些数据是可恢复的，所以小 H 决定帮助三叶。

由于某些原因，这些数据有依赖关系，复原一个数据需要复原它所依赖的数据。这些依赖关系形成一棵树。

对于一个数据，假设它有两个系数 a_i, b_i ，在复原这个数据时，因为还要保证没有被复原的数据的完整性，需要的代价为：

$$b_i \cdot \sum_{j \text{ 还没被复原}} a_j$$

小 H 认为自己一定可以在规定时间内复原出数据，但是她想捉弄一下三叶。因此小 H 想要让复原的时间尽可能长，也就是让上述代价尽量大。

现在你知道 n 个数据的系数和它们之间的依赖关系，请求出最大代价。

【形式化题意】

给定一棵 n 个点的树，对于所有 $2 \leq i \leq n$ ，它的父亲节点为 f_i ，每一个点有两个系数 a_i, b_i 。

你需要求出一个长度为 n 的排列，满足对于 $2 \leq i \leq n$ ， f_i 都在 i 出现前出现。

这个排列的代价为：

$$\sum_{i=1}^n \left(b_{p_i} \cdot \sum_{j=i+1}^n a_{p_j} \right)$$

求最大代价。

【输入格式】

从文件 `data.in` 中读入数据。

第一行一个整数 n ，表示需要复原的数据的个数。

第二行 $n-1$ 个整数，第 i 个数表示 f_{i+1} ，即复原第 $i+1$ 个数据需要先复原第 f_{i+1} 个数据。

接下来 n 行，每行两个整数 a_i, b_i ，表示第 i 个数据自身的两个系数。

【输出格式】

输出到文件 `data.out` 中。
 输出一行，表示最大的代价。

【样例 1 输入输出】

data1.in	data1.ans
4 1 1 2 0 0 3 1 5 1 4 1	14

【样例 1 解释】

可以按照 1,2,4,3 的方式选择。

【样例 2、3、4】

见选手目录下的 `data/data*.in` 和 `data/data*.ans`。

【数据范围和提示】

本题采用捆绑测试。

对于所有数据，满足 $1 \leq n \leq 3 \times 10^5, 1 \leq a_i, b_i \leq 5000, 1 \leq f_i < i$ 。

子任务见下表：

子任务编号	n	特殊性质	分值
1	≤ 20	无	15
2	≤ 1000		
3	$\leq 5 \times 10^4$		
4	$\leq 3 \times 10^5$	$f_i = i - 1$	10
5		$f_i = 1$	15
6		无	30

【后记】

自信的小 H 最后并没有恢复所有被清除的数据，因此这场模拟赛原本的 T1 丢失了，然后有了你们现在看到的这道题。

下落的小球(ball)

【题目背景】

宫水三叶制造了一台摇奖机。

这台摇奖机呈一棵树的形状，树上的每一个节点都有一个小球。这棵树的 1 号节点是这棵树的树根，也是摇奖机的最高处。

对于一个点上的小球，如果它的儿子节点中有空位，那么由于重力，小球会滚落到它有空位的儿子节点上，这样这个点就空了。

这是一道题的模板，三叶在翻自己写过的题目时发现了这个模板。

这是很早以前的事了，当时测试时某一题就是在这个背景下出的题目。但是那道题被小 H 一眼秒掉。

时隔多年，三叶决定再次使用这台摇奖机来出题。

【题目描述】

摇奖机可以用一棵大小为 n 的树来描述。

有一棵大小为 n 的树，其中 1 号点为根节点。记 1 号点是深度为 1，其他点的深度为自己父亲节点的深度 +1。

一开始树上每一个节点都有一个小球。

因为摇奖机是竖直放置的，所以深度浅的点比深度深的点的重力势能大。因此，如果存在两个相邻的点，深度浅的点上有小球，而深度深的点上没有小球，那么小球会从深度浅的点滚向深度深的点。一棵树上的球会按照上述方式一直滚动，直到找不到满足上述条件的两个相邻的点为止。

三叶给每一个叶子节点都安上了一个出口，表示可以在叶子节点将该节点上的球取出。

因为这是摇奖机，所以每一个出口取球都有一个次数限制。记 a_i 为第 i 个节点的限制，那么在第 i 个节点不能取球超过 a_i 次。

三叶把限制设置得很有规律，满足 $\sum_{i=1}^n a_i = n$ 。

每一次取完球后，该叶子节点上就没有小球了，此时摇奖机中的球就会因为重力滚动。等所有球都停止滚动后才能取下一次球。

三叶希望在限制次数内能把摇奖机中的 n 个球全部取出来，因此三叶需要你求出把小球全部取出的方案数。因为方案数很大，所有你只需要输出方案数对 $10^9 + 7$ 取模后的结果即可。

三叶用一个长度为 n 的序列 b 表示取球方案，表示第 i 次取球所在的叶子节点的编号。序列合法当且仅当你可以通过这次取球取完所有球，序列不同当且仅当存在某个位置不同。

【形式化题意】

一棵 n 个点的树，根为 1。对于 $2 \leq i \leq n$ ， f_i 为点 i 的父亲，树上每一个点都有一个标记值 s_i ，初始时所有 s_i 都为 1。

对点 x 进行操作表示把 x 的祖先中深度最小并且 $s_i = 1$ 的点 i 的标记值 s_i 赋值成 0。

你需要多少个不同的长度为 n 的序列 b ，满足在序列中数 i 出现了 a_i 次，并且对这棵树操作 n 次，每次操作 b_i ，每次操作前都满足 $s_{b_i} = 1$ 。

【输入格式】

从文件 `ball.in` 中读入数据。

第一行有一个整数 n 。

第二行有 $n - 1$ 个整数，第 i 个整数为 f_{i+1} ，即第 $i + 1$ 号点的父亲节点编号。

第三行共 n 个整数，第 i 个整数表示 a_i ，即第 i 个点的取球限制。

当 i 号点不是叶子节点时，保证 $a_i = 0$ 。

【输出格式】

输出到文件 `ball.out` 中。

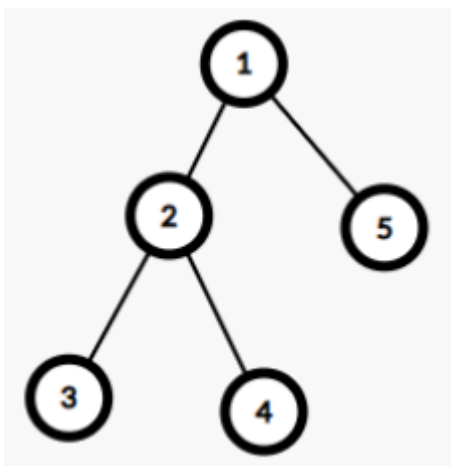
输出共一行，所有取球方案数对 $10^9 + 7$ 取模后的结果。

【样例 1 输入输出】

ball1.in	ball1.ans
5	16
1 2 2 1	
0 0 2 2 1	

【样例 1 解释】

如图，这是一棵大小为 5 的树。



树的叶子节点为 $\{3,4,5\}$ ，它们的限制次数为 $\{2,2,1\}$ 。

易知，第一次不可以在 5 号点取球，否则 5 号点次数用完，但 5 号点还有球。

因为 3,4 号点基本对称，我们假设第一次在 3 号点取球。

取完后 3 号点还有 1 次取球机会，而 4 号点还有两次取球机会。不考虑 5 号点，则一定要取 4 号点。

假设先取 4 号点，那么三个位置可以任意排序，方案为 $3! = 6$ 。

假设先取 5 号点，接下来一定取 4 号点，剩下两个球任意排序。

所以先取 3 号点的方案为 $(6 + 2) = 8$ 种。

那么总方案为 $2 \times 8 = 16$ 种。

当然，我们也可以使用列举法，方案如下：

$\{3,5,4,3,4\}, \{3,5,4,4,3\}, \{3,4,3,4,5\}, \{3,4,3,5,4\}, \{3,4,4,3,5\}, \{3,4,4,5,3\}$
 $\{3,4,5,3,4\}, \{3,4,5,4,3\}, \{4,5,3,3,4\}, \{4,5,3,4,3\}, \{4,3,3,4,5\}, \{4,3,3,5,4\}$
 $\{4,3,4,3,5\}, \{4,3,4,5,3\}, \{4,3,5,3,4\}, \{4,3,5,4,3\}$

上述方案共 16 种。

【样例 2、3】

见选手目录下的 ball/ball*.in 和 ball/ball*.ans。

【数据范围和提示】

本题采用捆绑测试。

对于所有数据，满足 $1 \leq n \leq 10^6, 1 \leq f_i < i$ 。

对于每个节点，满足 $1 \leq a_i \leq n, \sum_{i=1}^n a_i = n$ 。

子任务见下表：

子任务编号	n	特殊性质	分值
1	≤ 10	无	15
2	≤ 200		
3	$\leq 10^3$		
4	$\leq 10^6$	特殊性质 A	10
5		特殊性质 B	
6		特殊性质 C	
7		特殊性质 D	
8		无	15

特殊性质 A：保证树的最大深度不超过 3。

特殊性质 B：保证这棵树的根的度数为 1。对于所有的 $2 \leq i \leq n$ ，有且仅有一个点的度数为 3，其他点的度数都不超过 2。这里点的度数指的是一个点与之相连的边的数量。

特殊性质 C：保证整颗树是一棵满二叉树。换句话说，存在一个整数 k ，使得 $n = 2^k$ ，且对于所有的 $2 \leq i \leq n$ ，都有 $f_i = \lfloor \frac{i}{2} \rfloor$ 。

特殊性质 D：保证整棵树为二叉树。也就是说，对于每一个节点，最多有两个儿子。

消失的运算符(operator)

【题目描述】

宫水三叶在教妹妹四则运算。她出了这样一道题：给定一个长度为 n 的合法运算表达式，求该表达式的值。

为了降低题目难度，三叶规定数字只有 $1 \sim 9$ ，运算符只有加号和乘号。

为了规范题目，三叶对合法的表达式做了以下规定：

1. 数字 $1 \sim 9$ 是合法表达式。
2. 如果表达式 A 是合法表达式，那么 (A) 是合法表达式。
3. 如果表达式 A, B 是合法表达式，那么 $A + B, A * B$ 是合法表达式。
4. 以上三条规则可以构造出所有的合法表达式。

三叶把括号，数字，运算符等都做成了卡片，并摆出了表达式。

然而，不知道谁趁三叶不注意，把表达式中的运算符都取下来了。

三叶知道哪些位置有运算符，但是不知道每个运算符是什么。她只知道总共有 m 个运算符空位，而手上有 k 张加法的卡片和 $m - k$ 张乘法的卡片。

因此三叶想知道，对于所有可能的表达式，表达式的值的和是多少。

由于这个值很大，因此三叶只想知道其对 $10^9 + 7$ 取模后的结果。

【形式化题意】

给定一个长度为 n 的表达式，表达式合法，并且只出现括号，减号和数字 $1 \sim 9$ 。其中减号代表运算符的空位，设有 m 个。

给定 k ，你要把 k 个减号替换成加号，把 $m - k$ 个减号替换成乘号。求所有本质不同的替换方案表达式的和对 $10^9 + 7$ 取模后的值。

【输入格式】

从文件 `operator.in` 中读入数据。

第一行两个整数 n, k ，表示运算式长度和加号的个数。

第二行一个长度为 n 的字符串，表示运算式。

【输出格式】

输出到文件 `operator.out` 中。

输出一行，表示所有可能出现的表达式的和对 $10^9 + 7$ 取模后的结果。

【样例 1 输入输出】

operator1.in	operator1.ans
5 1 1-2-3	12

【样例 1 解释】

共有两个运算符，其中一个为加号。
共有两个合法的表达式：

1. $1 + 2 * 3 = 7$ 。

2. $1 * 2 + 3 = 5$ 。

因此答案为 $7 + 5 = 12$

【样例 2、3】

见选手目录下的 `operator/operator*.in` 和 `operator/operator*.ans`。

【数据范围和提示】

本题采用捆绑测试，每个子任务 20 分。

对于所有数据，满足 $1 \leq n \leq 10^5, 0 \leq m \leq 2500, 0 \leq k \leq m$ 。

保证给出的表达式合法。

子任务见下表：

子任务编号	m	特殊性质
1	≤ 20	无
2	≤ 300	
3	≤ 2500	特殊性质 A
4		特殊性质 B
5		无

特殊性质 A：满足 $k = 0$ 。

特殊性质 B：保证表达式中不出现括号。

古老的序列问题(sequence)

【题目背景】

临近 NOIP，机房里的所有同学都在努力刷题。

老师觉得 NOIP 前，冲刺难题固然重要，但是基础也需要好好练习。因此，老师经常布置一些简单的题目让大家完成。

但是，大家都对简单题都不屑一顾。

某天集训结束，老师在教室前面的黑板上留下一道题：

给定长度为 n 的序列 s ，求 s 的所有连续子序列的最大值之和。

同学们议论纷纷，认为这道题非常简单。

宫水三叶也在集训，但是她认为这道题挺难的。所以三叶把这道题抄回家，并通过努力解决了这道题。

第二天，三叶来到学校。但是，黑板上的题目不知道为什么变了。

给定长度为 n 的序列 s ，求 s 的所有连续子序列的最大值和最小值的乘积之和。

老师走进的教室，并询问谁改了这道题，小 H 举起了手。

老师让小 H 解决她改后的这道题，小 H 走上讲台，并拿起了笔。

她在题目描述的某个位置又加上了四个字……

【题目描述】

你有一个长度为 n 的整数序列 s ，其中第 i 个数为 s_i 。

你要回答 m 个询问，每一次询问将给出 L, R ，你需要求出下面这个式子的值：

$$\left(\sum_{l=L}^R \sum_{r=l}^R \text{cost}(l, r) \right) \pmod{10^9 + 7}$$

其中 $\text{cost}(l, r)$ 为 $[l, r]$ 中最大值和最小值的乘积，即：

$$\text{cost}(l, r) = (\max_{i=l}^r s_i) \cdot (\min_{i=l}^r s_i)$$

【输入格式】

从文件 `sequence.in` 中读入数据。

第一行两个整数 n, m ，表示 s 的长度和询问的个数。

第二行 n 个整数，第 i 个整数为 s_i 。

接下来 m 行，每行两个整数 L, R ，表示一个询问。

【输出格式】

输出到文件 `sequence.out` 中。

输出 m 行，第 i 行为第 i 个询问的答案。

【样例 1 输入输出】

sequence1.in	sequence1.ans
5 2	77
2 4 1 5 3	60
1 4	
3 5	

【样例 1 解释】

以下为所有区间的最小值：

min	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$
$l = 1$	2	2	1	1	1
$l = 2$	—	4	1	1	1
$l = 3$	—	—	1	1	1
$l = 4$	—	—	—	5	3
$l = 5$	—	—	—	—	3

以下为所有区间的最大值：

max	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$
$l = 1$	2	4	4	5	5
$l = 2$	—	4	4	5	5
$l = 3$	—	—	1	5	5
$l = 4$	—	—	—	5	5
$l = 5$	—	—	—	—	3

以下为所有区间的贡献值：

$cost$	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$
$l = 1$	4	8	4	5	5
$l = 2$	—	16	4	5	5
$l = 3$	—	—	1	5	5
$l = 4$	—	—	—	25	15
$l = 5$	—	—	—	—	9

通过查表，容易得知：

当 $L = 1, R = 4$ 是，贡献和为 77。

当 $L = 3, R = 5$ 是，贡献和为 60。

【样例 2】

见选手目录下的 `sequence/sequence*.in` 和 `sequence/sequence*.ans`。

【数据范围和提示】

本题采用捆绑测试，每个子任务 10 分。

对于所有数据，满足 $1 \leq n, m \leq 10^5, 1 \leq s_i \leq 10^9, 1 \leq L \leq R \leq n$ 。

子任务见下表：

子任务编号	n	m	特殊性质
1	≤ 100	≤ 100	无
2	≤ 2000	≤ 2000	
3	$\leq 5 \times 10^4$	$\leq 5 \times 10^4$	
4	$\leq 10^5$	$= 1$	s 随机生成
5		$\leq 10^5$	$s_i \leq 5$
6			$s_i \leq s_{i+1}$
7			$L = 1$
8			s 随机生成
9			无
10			无

s 随机生成：表示对于 $1 \leq i \leq n$ ， s_i 在 $[1, 10^9]$ 中等概率随机。