BDFZ NOIP 2020

第二试

题目名称	生活在树上	三千世界	大过滤器	碎梦
题目类型	传统型	传统型	传统型	传统型
可执行文件名	tree.exe	thousands.exe	filter.exe	broken.exe
输入文件名	tree.in	thousands.in	filter.in	broken.in
输出文件名	tree.out	thousands.out	filter.out	broken.out
每个测试点时限	1.0 秒	1.0 秒	0.5 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点/包数目	10	10	7	20
测试点是否等分	是	是	否	是

提交源程序文件名

对于 C++ 语言	tree.cpp	thousands.cpp	filter.cpp	broken.cpp

编译选项

对于 C++ 语言	-O2 -std=c++11 -m32 -Wl,--stack=536870912

注意事项

1. 选手提交的源文件无需建子文件夹。
2. 若无特殊说明，输入文件中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。
3. 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。
4. 程序可使用的栈空间大小与该题内存空间限制一致。
5. 评测时采用的机器配置为：Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz，内存 8GB。上述时限以此配置为准。
6. 编译选项中的 -m32 命令意为仅编译在 32 位，64 位系统均可运行的可执行文件，因此选手程序中无法使用 128 位整型。

生活在树上（tree）

【题目描述】

小忆和小艾生活在树上。这颗树 T 有 n 个节点，由 $n-1$ 条边连接。现在树上有一个排列 p，每次小艾可以选择一条边 $(u, v) \in T$，将 p_u 与 p_v 交换，小艾的任务是将排列完成排序。为了估算自己至少要交换多少次，小艾找小忆请教，小忆经过思考，写下了这个式子：

$$\frac{1}{2}\sum_{i=1}^{n} \operatorname{dist}(i, p_i)$$

小艾经过尝试，发现很巧的是，当前的这个排列刚好达到了小忆给出的下界！他想考考你，你能不能给出一个达到这一下界的排序方案呢？特别地，她还希望你给出的方案的**字典序最小**。我们认为树上的边是从 1 至 $n-1$ 编号的，方案的字典序是由依次比较操作的边的编号决定的。

【输入格式】

从文件 ***tree.in*** 中读入数据。

第一行输入一个正整数 n，表示树的节点数。

接下来 $n-1$ 行每行输入两个正整数 u_i, v_i，表示第 i 条边（$1 \le i \le n-1$）。

接下来一行 n 个正整数，表示排列 p。

【输出格式】

输出到文件 ***tree.out*** 中。

输出一行。按顺序输出字典序最小的解中，操作的边的编号。

【样例 1 输入】

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

【样例 1 输出】

```
2 1 3 4 2
```

【样例 1 解释】

初始序列为 $2,1,5,3,4$。接下来的 5 次操作过程中，序列变为：

- $2,\underline{5},\underline{1},3,4$
- $2,\underline{4},1,3,\underline{5}$
- $2,\underline{3},1,\underline{4},5$
- $\underline{1},3,\underline{2},4,5$
- $1,\underline{2},\underline{3},4,5$

【样例 2】

见选手目录下的 ***tree/tree2.in*** 与 ***tree/tree2.ans***。

【子任务】

对于 100% 的数据，保证 $1 \le n \le 10^3$，且给出的排列可以在 $\frac{1}{2}\sum_{i=1}^{n} \operatorname{dist}(i, p_i)$ 次操作内完成排序。

测试点	n	特殊限制
1,2	$=5$	
3,4	$=30$	
5	$=10^2$	
6		A0
7		A1
8	$=10^3$	B0
9		B1
10		

特殊限制：

- A 表示边集为 $\{(1,2),(2,3),\ldots,(n-1,n)\}$。
- B 表示边集为 $\{(1,2),(1,3),\ldots,(1,n)\}$。
- 0 表示保证边按照前述所给的顺序给出。
- 1 表示可能以任何一个顺序给出。

三千世界（thousands）

【题目背景】

对于一颗树上的一些路径组成的集合 S，定义 $f(S)$ 为最大的子集 $T \subseteq S$ 的大小，满足 T 内的路径点两两不交。我们认为点对 (x, y) 表示了一条路径。

对于全体路径 $P = \{(x, y) \mid 1 \le x, y \le n\}$，试求出

$$\sum_{S \subseteq P} f(S)$$

即你需要对这 2^{n^2} 种集合的 f 值求和。你只需要输出对 998244353 取模的结果。

【输入格式】

从文件 ***thousands.in*** 中读入数据。

第一行输入一个正整数 n，表示树的节点数量。

接下来 $n-1$ 行，每行输入两个正整数 u, v 表示树上的一条边。

【输出格式】

输出到文件 ***thousands.out*** 中。

输出一行一个整数，表示答案对 998244353 取模的结果。

【样例 1 输入】

```
2
1 2
```

【样例 1 输出】

```
19
```

【样例 1 解释】

f 值为 0 的有 $\varnothing$ 这一个。f 值为 2 的必须含有 $(1,1)$ 和 $(2,2)$，而 $(1,2)$ 和 $(2,1)$ 任取，有 4 种。剩下的 11 种集合的 f 值均为 1，因此答案为 $0 \times 1 + 1 \times 11 + 2 \times 4 = 19$。

【样例 2 输入】

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

【样例 2 输出】

```
103767551
```

【子任务】

对于 100% 的数据，保证 $1 \le n \le 5,000$。

<table>
<tr><th>测试点</th><th>n</th><th>特殊限制</th></tr>
<tr><td>1</td><td>$= 2$</td><td rowspan="5"></td></tr>
<tr><td>2</td><td>$= 3$</td></tr>
<tr><td>3</td><td>$= 4$</td></tr>
<tr><td>4</td><td>$= 5$</td></tr>
<tr><td>5,6</td><td>$= 200$</td></tr>
<tr><td>7</td><td rowspan="3">$= 5,000$</td><td>A</td></tr>
<tr><td>8</td><td>B</td></tr>
<tr><td>9,10</td><td></td></tr>
</table>

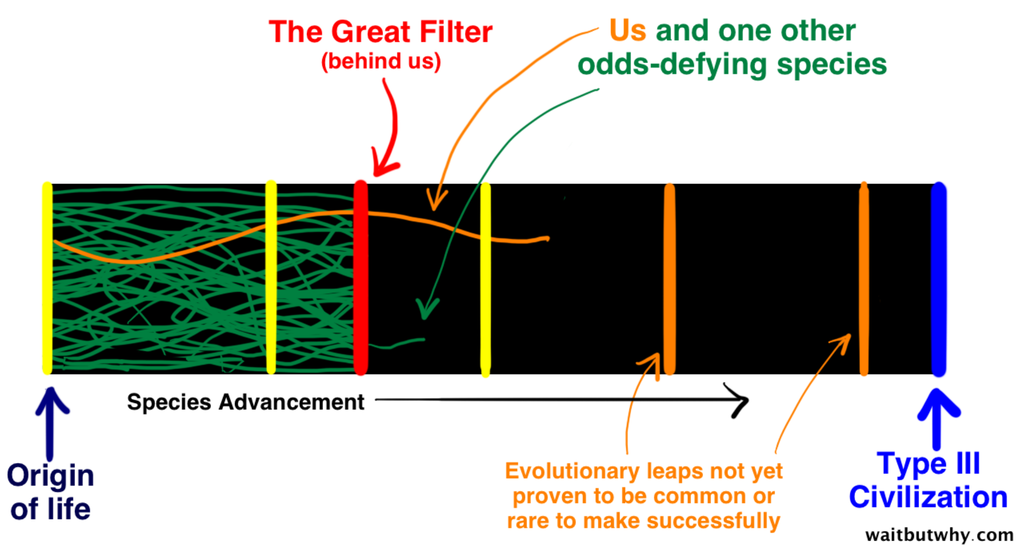
特殊限制：

- A 表示边集为 $\{(1,2),(2,3),\ldots,(n-1,n)\}$。
- B 表示边集为 $\{(1,2),(1,3),\ldots,(1,n)\}$。

大过滤器（filter）

【题目背景】

大过滤器理论（The Great Filter）认为，文明在发展过程中存在若干个重要的划分阶段，各阶段之间存在着极难跨越的沟壑，以至于达到最终可以实现星际殖民阶段的文明少之又少。这一理论也被认为是费米悖论的一种解释。



【题目描述】

在本题中，我们认为文明存在 n 个级别，而这 n 个级别又被划分为 k 个阶段。具体地，我们有数组 $L_0, L_1, \ldots, L_k$，满足 $0 = L_0 < L_1 < \cdots < L_k = n$，其中第 $L_{j-1}+1$ 到第 L_j 个文明级别被认为是处于阶段 j 的。

我们认为一张有向图 $G = (V, E)$ 刻画了文明可以通过什么手段达到最终级别。若 $(x, y) \in E$，则说明处于 x 级别的文明可以尝试进步到 y 级别（注意这里并不保证 $x < y$！）特别地，由于大过滤器的存在，设 x 是 j 阶段的文明，那么 y 只可能处于 j 阶段或者 $j+1$ 阶段，如果 y 也属于 j 阶段，那么我们认为这是一次“常规进步”，否则 y 属于 $j+1$ 阶段，我们认为这是一次“危险进步”。

我们认为现在人类文明所处的级别为 1 级别，我们的目标是达到 i 级别，我们需要规划一个进步方案。方案可以表示为 1 到 n 的一条路径，我们如此定义一种方案的困难程度：设计数器初始有 $s = 0$，我们按顺序考虑这条路径，每次发生一次“常规进步”，那么 $s \leftarrow s+1$，每次发生一次“危险进步”，那么 $s \leftarrow s \times 2$；最后的 s 值就是该进步方案的困难程度。

对于每个 $1 \le i \le n$，请你判断是否存在一种从 1 级别进步至 i 级别的方案，如果存在，那么请规划一种方案使得困难程度最小。

【输入格式】

从文件 ***filter.in*** 中读入数据。

第一行输入三个正整数 n, m, k，表示文明的级别数量，图的边数，以及文明的阶段数。

接下来一行输入 $k-1$ 个正整数，表示 $L_1, \ldots, L_{k-1}$，如题意所示。

接下来 m 行每行输入两个正整数 x, y 表示一条边。

【输出格式】

输出到文件 ***filter.out*** 中。

输出共 n 行，第 i 行一个整数，表示从 1 级别进化到 i 级别的最小困难程度。由于这个数很大，你只需要输出其 $\bmod 998244353$ 的结果即可。如果无法进化到 i 级别，输出 -1。

【样例 1 输入】

```
6 6 2
3
1 2
2 3
3 4
4 5
5 6
2 6
```

【样例 1 输出】

```
0
1
2
4
5
2
```

【样例 2】

见选手目录下的 ***filter/filter2.in*** 与 ***filter/filter2.ans***。

【样例 2 解释】

注意取模。

【样例 3】

见选手目录下的 ***filter/filter3.in*** 与 ***filter/filter3.ans***。

【子任务】

对于 100% 的数据，保证 $2 \le k \le n \le 3 \times 10^5, 1 \le m \le 5 \times 10^5, 1 \le x, y \le n$。

<table>
<tr><th>测试点</th><th>分值</th><th>$n \le$</th><th>$m \le$</th><th>$k \le$</th></tr>
<tr><td>1</td><td>10</td><td>10^2</td><td>200</td><td rowspan="3">40</td></tr>
<tr><td>2</td><td>15</td><td>10^5</td><td>2×10^5</td></tr>
<tr><td>3</td><td>10</td><td>3×10^5</td><td>5×10^5</td></tr>
<tr><td>4</td><td>20</td><td>500</td><td>10^3</td><td rowspan="4">n</td></tr>
<tr><td>5</td><td>20</td><td>3×10^4</td><td>6×10^4</td></tr>
<tr><td>6</td><td>15</td><td>10^5</td><td>2×10^5</td></tr>
<tr><td>7</td><td>10</td><td>3×10^5</td><td>5×10^5</td></tr>
</table>

【提示】

本题输入文件在 10 MB 以内，输出文件在 5 MB 以内，请使用较快的输入输出方式。

碎梦（broken）

【题目背景】

「从我又小又拥挤的房间」

「从我堆满了贪念的床边」

「飞向总有容身处的世界」

「却碎裂 却碎裂」

——「碎梦」COP，洛天依

多年以后，算法竞赛的一切对于忆哀来说是什么呢？一个光辉而没有结局的梦罢了。

永恒的梦，失落的梦，无望的梦，破碎的梦。

仿佛是想到了什么，她又从纸箱里拿出落了几层灰的笔记本，好不容易找出旧时代的充电插头。随着散热风扇的嗡鸣声，一个尘封的世界再次展现在她眼前，还觉得自己只要稍微看看问题，就有可能抽丝剥茧理清问题的思路，那只是逞强罢了。

“第二类斯特林数怎么求来着？”这个问题突然从她脑中浮现。

是啊，她曾经深深扎入各种组合计算中流连忘返，但如今已经连这些基础也不甚记得了。

【题目描述】

第二类斯特林数，是说将 n 个数划分进 m 个非空集合的方案数。我们接下来记为 $f(n,m)$。

忆哀的程序使用的是一个很朴素的递推式：$f(n,m) = f(n-1,m-1) + mf(n-1,m)$，初值为 $f(0,0) = 1, f(0,m) = 0 (m \neq 0)$，这个递推式的意义是不难解释的：要么第 n 个元素是自成一个集合，要么就将其分配给已有的 m 个集合之一。

忆哀的程序是这样的：

```
for (int i = 1; i <= n; ++i)
  for (int j = 0; j <= min(i, m); ++j)
    f[i][j] = (f[i - 1][j - 1] +
      j * (long long)f[i - 1][j]) % 998244353;
```

（你可以认为数组越界的部分值为 0，并且电脑开的下 $(n+1)\times(m+1)$ 的数组）

最后忆哀的程序理应输出 $f(n,m) \bmod 998244353$，但出了一个问题：由于种种原因，在计算完 $f(x,y)$ 后，内存的写入产生了意外，在内存中 $f(x,y)$ 被赋值为了一个数 z，这样的事件对于不同的 (x,y) 总共发生了 k 次。

请你输出经过了这 k 次意外后，实际上忆哀的程序给出的 $f(n,m) \bmod 998244353$ 为多少。

【输入格式】

从文件 ***broken.in*** 中读入数据。

第一行输入三个整数 n, m, k，意义如题目描述所示。

接下来 k 行每行输入三个数 x_i, y_i, z_i，表示 $f(x_i, y_i)$ 在计算完成后实际写入的值为 z_i。

【输出格式】

输出到文件 ***broken.out*** 中。

输出一个整数，表示计算得到的实际 $f(n, m)$ 结果。

【样例 1 输入】

```
5 3 1
1 0 1
```

【样例 1 输出】

```
31
```

【样例 2 输入】

```
1000 100 0
```

【样例 2 输出】

```
958221900
```

【样例 3】

见选手目录下的 ***broken/broken3.in*** 与 ***broken/broken3.ans***。

【子任务】

对于 100% 的数据，保证 $1 \le x_i \le n \le 9 \times 10^8, 0 \le y_i \le m \le \min(n, 10^5), 0 \le k \le 20, 0 \le y_i \le x_i, 0 \le z_i < 998244353, (x_i < x_{i+1}) \vee (x_i = x_{i+1} \wedge y_i < y_{i+1})$。

<table>
<tr><th>测试点</th><th>$n \le$</th><th>$m \le$</th><th>$k =$</th></tr>
<tr><td>1,2,3,4,5,6</td><td>10^3</td><td>500</td><td rowspan="2">20</td></tr>
<tr><td>7,8,9</td><td rowspan="6">9×10^8</td><td>10</td></tr>
<tr><td>10,11</td><td rowspan="2">10^2</td><td>0</td></tr>
<tr><td>12,13,14</td><td rowspan="2">20</td></tr>
<tr><td>15,16,17</td><td>500</td></tr>
<tr><td>18</td><td rowspan="2">10^5</td><td>0</td></tr>
<tr><td>19,20</td><td>20</td></tr>
</table>