

NOI2022 省选

DAY1

时间：2023 年 4 月 1 日 08:30 ~ 2022 年 4 月 1 日 13:00

题目名称	火车站	城市建造	人员调度
题目类型	传统型	传统型	传统型
目录	station	cities	transfer
可执行文件名	station	cities	transfer
输入文件名	station.in	cities.in	transfer.in
输出文件名	station.out	cities.out	transfer.out
每个测试点时限	1.0 秒	1.0 秒	5.0 秒
内存限制	512 MiB	512 MiB	512 MiB
测试点数目	10	20	50
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	station.cpp	cities.cpp	transfer.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。
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：Intel(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。

火车站 (station)

【题目描述】

有 n 个火车站排成一条直线，从 1 到 n 编号。一共有 m 条火车轨道，每条轨道覆盖一段火车站区间 $[l_i, r_i]$ 。

对于一个被多条火车轨道覆盖的火车站，火车在经过这里的时候，可以在此处改变轨道。但是火车无法掉头，只能朝着一个方向运行（即只能一直往 1 的方向开或者一直往 n 的方向开）。

小 A 从火车站 x 出发，即搭上了经过 x 的任意一列火车（这列火车也可能是从车站 x 出发）。这列火车可能行驶在火车站 x 所处的任一条轨道上，其运行方向既可能是往 1 的方向开，也可能是往 n 的方向开。小 A 上车后就开始昏睡，直到乘坐的火车到达某条线路的终点站停下，他才醒过来。问小 A 最后可能到达的车站。

注意：火车应运行至少一个车站，且火车切换轨道后不会立刻停下来，而是会继续沿着当前轨道前进。

【输入格式】

从文件 *station.in* 中读入数据。

输入的第一行包含三个正整数 n, m, x ，分别表示火车站的数量，火车轨道的数量以及小 A 初始的起点。

接下来 m 行，每行包含两个正整数 l_i, r_i ，表示一条火车轨道运行的区间。

【输出格式】

输出到文件 *station.out* 中。

输出一行，包含若干个用单个空格分隔的正整数，表示小 A 最后可能到达的车站，按照车站编号升序排序输出。

【样例 1 输入】

```
1 7 5 4
2 3 4
3 4 6
4 1 3
5 5 7
6 4 6
```

【样例 1 输出】

```
1 1 3 6 7
```

【样例 1 解释】

火车从车站 4 出发，沿着第一条轨道可以运行到终点 3，也可以接着沿第三条轨道运行到终点 1。

火车从车站 4 出发，沿着第二条轨道可以运行到终点 6，也可以在车站 5 换到第四条轨道运行到终点 7。

所以最终按顺序输出 1, 3, 6, 7。

【样例 2】

见选手目录下的 *station/station2.in* 与 *station/station2.ans*。

【样例 3】

见选手目录下的 *station/station3.in* 与 *station/station3.ans*。

【样例 4】

见选手目录下的 *station/station4.in* 与 *station/station4.ans*。

【数据范围】

对于所有的数据，保证 $1 \leq n, m \leq 2 \times 10^5$ ； $1 \leq x \leq n$ ； $1 \leq l_i < r_i \leq n$ 。

测试点	$n, m \leq$	特殊性质
1	50	无
2		
3		
4		
5		
6	2×10^5	A
7		无
8		
9		
10		

特殊性质 A：保证 $x = 1$ 。

城市建造 (cities)

【题目描述】

在这个国度里面有 n 座城市，一开始城市之间修有若干条双向道路，导致这些城市形成了 $t \geq 2$ 个连通块，特别的，这些连通块之间两两大小差的绝对值不超过 $0 \leq k \leq 1$ 。为了方便城市建设与发展， n 座城市中的某 t 座城市在这 t 座城市之间额外修建了至少一条双向道路，使得所有城市连通。

现在已经知道额外修建后的所有道路，你需要算出有哪些双向道路集合 E' ，满足这些道路有可能是后来额外修建的，请输出答案对 998,244,353 取模的结果。

即给定一张 n 个点 m 条边的无向连通图 $G = (V, E)$ ，询问有多少该图的子图 $G' = (V', E')$ ，满足 $E' \neq \emptyset$ 且 $G - E'$ 中恰好有 $|V'|$ 个连通块，且任意两个连通块大小之差不超过 k ，保证 $0 \leq k \leq 1$ ，请输出答案对 998,244,353 取模的结果。

【输入格式】

从文件 *cities.in* 中读入数据。

输入的第一行包含三个正整数 n, m, k ，分别表示城市数、修建后的道路数以及任意两个连通块大小之差的上限。

接下来 m 行每行包含两个正整数 u, v ，表示城市 u 和 v 之间存在一条双向道路，保证 $u \neq v$ 。

【输出格式】

输出到文件 *cities.out* 中。

输出一个数表示答案对 998,244,353 取模后的结果。

【样例 1 输入】

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

【样例 1 输出】

```
1 2
```

【样例 1 解释】

有以下两种情况：

- 本来只有 $(3, 4)$ 这一条道路，此时有三个连通块，分别为 $\{1\}, \{2\}, \{3, 4\}$ ；后来城市 1, 2, 3 决定在他们三座城市中额外修建了 $(1, 2), (2, 3), (1, 3)$ 这三条道路，使得所有城市连通。
- 本来没有任何道路，此时有四个连通块，分别为 $\{1\}, \{2\}, \{3\}, \{4\}$ ；后来城市 1, 2, 3, 4 决定在他们四座城市中额外修建了 $(1, 2), (2, 3), (1, 3), (3, 4)$ 这四条道路，使得所有城市连通。

【样例 2】

见选手目录下的 *cities/cities2.in* 与 *cities/cities2.ans*。

【样例 3】

见选手目录下的 *cities/cities3.in* 与 *cities/cities3.ans*。

【样例 4】

见选手目录下的 *cities/cities4.in* 与 *cities/cities4.ans*。

【数据范围】

对于所有的数据，保证： $3 \leq n \leq 10^5$ ； $n - 1 \leq m \leq 2 \times 10^5$ ； $0 \leq k \leq 1$ 。

测试点	n	m	k
1, 2	≤ 15	≤ 20	$= 0$
3 ~ 5	≤ 20	≤ 50	$= 1$
6, 7	≤ 200	≤ 300	$= 0$
8, 9	$\leq 2,000$	$= n - 1$	$= 1$
10, 11		$\leq 3,000$	$= 0$
12, 13			$= 1$
14, 15	$\leq 10^5$	$= n - 1$	$= 1$
16, 17		$\leq 2 \times 10^5$	$= 0$
18 ~ 20			$= 1$

人员调度 (transfer)

【题目描述】

众所周知，一个公司的 n 个部门可以组织成一个树形结构。形式化地，假设这些部门依次编号为 $1, \dots, n$ ，那么除了 1 号部门以外，第 $i \in [2, n]$ 个部门有且仅有一个上级部门 $p_i \in [1, i-1]$ 。这样，这家公司的 n 个部门可以视为一个以 1 为根的树。如果 i 是 j 子树中的点，那么称部门 i 是部门 j 的子部门。

该公司初始时有 k 名优秀员工，编号依次为 $1 \dots k$ 。第 i 名优秀员工初始时在第 x_i 个部门工作，并且其有一个能力值 $v_i > 0$ 。

为了最大化公司的运作效率，公司老板 $0/\backslash/\backslash G$ 决定进行一些人员调动。具体来说，可以将编号为 i 的优秀员工调动到 x_i 的一个子部门，或者不调度（此时该员工在 x_i 部门）。随后，优秀员工们会在其所在的部门竞选部门领导——能力值最高者将担任这一职位，并给公司带来等同于其能力值的贡献。如果一个部门一个优秀员工也没有，那么就无法选出部门领导，从而对公司的贡献将是 0。此时，公司的业绩被定义为公司各部门的贡献之和。

公司老板 $0/\backslash/\backslash G$ 自然想知道，该如何进行人员调动，使公司的业绩最大？

这当然难不倒他，然而，公司优秀员工的数量也会发生变化；具体来说，会依次发生 m 个事件，每个事件形如：

1 $x \ v$ ：先令 $k = k + 1$ ，然后新增一位编号为 k 、初始部门为 x 、能力值为 v 的优秀员工；

2 id ：编号为 id 的优秀员工将被辞退。

公司老板 $0/\backslash/\backslash G$ 希望你能在最开始和每个事件发生后，告诉他公司的业绩最大可能是多少？

注意，每次人员调动都是独立的，也就是每次计算公司的最大可能业绩时，每个优秀员工都会回到其所在的初始部门。

【输入格式】

从文件 **transfer.in** 中读入数据。

输入的第一行包含一个正整数 sid ，表示该测试点对应的数据范围以及特殊性质，详见后表；

输入的第二行包含三个整数 n, k, m ，分别表示部门数，初始优秀员工数和事件数。

输入的第三行包含 $n - 1$ 个正整数 p_2, \dots, p_n ，表示每个部门的上级部门。

接下来 k 行，每行包含两个正整数 x_i, v_i ，表示优秀员工的初始部门和能力值。

接下来 m 行，每行形如 1 $x \ v$ 或 2 id 表示一次事件。

【输出格式】

输出到文件 *transfer.out* 中。

输出一行包含 $m + 1$ 个由单个空格隔开的非负整数，依次表示最开始和每个事件发生后，公司的业绩可能的最大值。

【样例 1 输入】

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

【样例 1 输出】

```
1 4 5
```

【样例 2】

见选手目录下的 *transfer/transfer2.in* 与 *transfer/transfer2.ans*。

【样例 3】

见选手目录下的 *transfer/transfer3.in* 与 *transfer/transfer3.ans*。

【样例 4】

见选手目录下的 *transfer/transfer4.in* 与 *transfer/transfer4.ans*。

【样例 5】

见选手目录下的 *transfer/transfer5.in* 与 *transfer/transfer5.ans*。

【样例 6】

见选手目录下的 *transfer/transfer6.in* 与 *transfer/transfer6.ans*。

【样例 7】

见选手目录下的 *transfer/transfer7.in* 与 *transfer/transfer7.ans*。

【样例 8】

见选手目录下的 *transfer/transfer8.in* 与 *transfer/transfer8.ans*。

【样例 9】

见选手目录下的 *transfer/transfer9.in* 与 *transfer/transfer9.ans*。

【样例 10】

见选手目录下的 *transfer/transfer10.in* 与 *transfer/transfer10.ans*。

【样例 11】

见选手目录下的 *transfer/transfer11.in* 与 *transfer/transfer11.ans*。

【样例 12】

见选手目录下的 *transfer/transfer12.in* 与 *transfer/transfer12.ans*。

【样例 13】

见选手目录下的 *transfer/transfer13.in* 与 *transfer/transfer13.ans*。

【样例 14】

见选手目录下的 *transfer/transfer14.in* 与 *transfer/transfer14.ans*。

【样例 15】

见选手目录下的 *transfer/transfer15.in* 与 *transfer/transfer15.ans*。

【数据范围】

对于所有的数据, 保证: $1 \leq sid \leq 15$; $1 \leq n, k \leq 10^5$; $0 \leq m \leq 10^5$; $1 \leq p_i < i$;
 $1 \leq x_i, x \leq n$; $1 \leq v_i, v \leq 10^5$;

对于事件 2, 保证: $1 \leq id \leq k$ 且编号为 id 的员工在此事件发生仍在工作。

测试点编号	sid	$n \leq$	$k \leq$	$m \leq$	特殊性质
1	1	6	6	6	无
2,3	2	9			
4,5	3	16	66	66	
6 ~ 8	4	66		0	
9 ~ 11	5	2, 333	2, 333		
12 ~ 14	6	10^5	10^5		B
15 ~ 18	7			无	
19 ~ 21	8	2, 333	2, 333	2, 333	A
22 ~ 24	9	10^5	10^5	10^5	AB
25 ~ 28	10				A
29 ~ 31	11	2, 333	2, 333	2, 333	无
32 ~ 34	12	10^5	10^5	10^5	C
35 ~ 38	13				B
39 ~ 44	14	66, 666	66, 666	66, 666	无
45 ~ 50	15	10^5	10^5	10^5	

特殊性质 A: 无事件 2;

特殊性质 B: $p_i = i - 1$;

特殊性质 C: $v_i = v = 1$;