

NOIP 2024 模拟赛

Fly Wang

题目名称	游戏	路径	回环	发绳
题目类型	传统题	传统题	传统题	传统题
目录	number	path	tree	rope
可执行文件名	number	path	tree	rope
输入文件名	number.in	path.in	tree.in	rope.in
输出文件名	number.out	path.out	tree.out	rope.out
每个测试点时限	1.0 秒	2.0 秒	2.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
子任务数目	6	6	7	10
子任务是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	number.cpp	path.cpp	tree.cpp	rope.cpp
-----------	------------	----------	----------	----------

编译选项

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

注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- 选手的源代码文件统一放置于每位选手的子目录下，不再对每个题单独建立子目录。
- 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`。
- 选手若违反以上三点，造成的后果统一不予受理。
- 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）。
- 程序可使用的栈内存空间限制与题目的内存限制一致。
- 只提供 Windows 格式附加样例文件。
- 评测时限根据 Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 制订，若有异常可合理调整。

游戏 (number)

【题目描述】

小 W 进入了一个游戏，在游戏中他扮演了救世英雄，需要打败一个怪物，其初始血量为 s 。

游戏分为 n 个回合，每个回合他可以选择是否攻击怪物。

若选择不攻击怪物，怪物则会恢复 a 点血量；若选择攻击怪物，则要求其血量 $\geq b$ ，可以使其减少 b 点血量。

任意时刻怪物血量不能为负，求 n 个回合后，怪物的最低血量。

【输入格式】

从文件 `number.in` 中读入数据。单个测试点含有 T 组测试数据。

第一行一个整数 T ，表示测试数据组数。

第 $2 \sim T + 1$ 行每行 4 个整数 n, a, b, s ，含义如题述。

【输出格式】

输出到文件 `number.out` 中。

输出 T 行每行一个整数，表述单组测试数据的答案。

【样例输入 1】

```
1 2
2 5 1 3 7
3 8 2 3 2
```

【样例输出 1】

```
1 0
2 3
```

【样例解释 1】

我们用 $+$ 表示怪物恢复 a 点血量， $-$ 表示小 W 攻击怪物使其减少 b 点血量。
对于第一组测试数据，可以使用 $-++-$ 使其最后变成 0 点血量。
对于第二组测试数据，可以使用 $++-+-$ 使其最后变成 3 点血量。

【样例 2】

见选手目录下的 `number2.in/number2.ans`。
该样例符合 *Subtask 3* 的限制。

【样例 3】

见选手目录下的 `number3.in/number3.ans`。
该样例符合 *Subtask 6* 的限制。

【数据范围】

对于 100% 的数据，有 $1 \leq T \leq 10^5, n, a, b, s \leq 10^9$ 。

子任务编号	$T \leq$	$n \leq$	$a, b, s \leq$	分值
1	10	10	10	15
2	10	100	100	10
3	10	5000	5000	20
4	10	10^9	5000	15
5	10^5	5000	10^9	15
6	10^5	10^9	10^9	25

路径 (path)

【题目描述】

现在有一个无向图，存在 n 个点， m 条边且无自环和重边，每个点有点权 v_i 。另外，我们有一个长度为 s 的序列 b_1, b_2, \dots, b_s ，其中 $1 \leq b_i \leq n$ 。小 Y 从 1 号点出发到 x 号点，将经过的路径记作一个序列 p_1, p_2, \dots, p_k ，问 $\{b_i\}$ 前缀中满足是任意 $\{v_{p_i}\}$ 的子序列的最长长度。 $x \in [1, n]$ 都需要求出答案。

形式化的，对所有 $x \in [2, n]$ 求 $\{c_i\}$ 的最长长度 ans ，使得满足以下条件：

- 1. $c_i = b_i (i \in [1, ans])$
- 2. $\{p_i\}$ 满足 $p_1 = 1, p_k = x$ 且 (p_i, p_{i+1}) 有边相连。
- 3. 对于任意 $\{p_i\}$ 都 $\exists \{z_i\}$ 满足 $1 \leq z_1 < z_2 \dots z_{ans} \leq s$ 且 $c_i = v_{p_{z_i}}$

【输入格式】

从文件 `path.in` 中读入数据。
第一行三个整数 n, m, s ，表示无向图的点数和边数，以及 b 序列的长度。
第二行 n 个整数 v_1, v_2, \dots, v_n 表示每个点的点权。
第二行 s 个整数 b_1, b_2, \dots, b_s 表示序列 $\{b_i\}$ 。
第 4 ~ $m + 3$ 行每行两个整数 u_i, v_i ，表示 (u_i, v_i) 之间有一条无向边相连。

【输出格式】

输出一行 n 个整数，第 i 个整数表示 $x = i$ 时的答案。

【样例输入 1】

```
1 5 5 3
2 3 2 1 4 2
3 2 1 3
4 1 2
5 1 3
6 2 4
7 3 4
8 4 5
```

【样例输出 1】

1

0 1 0 0 1

【样例解释 1】

当 $x = 2$ ，有两种不走重复点的路径 $\{1, 2\}$ 和 $\{1, 3, 4, 2\}$ ，对应的权值序列为 $\{3, 2\}$ 和 $\{3, 1, 4, 2\}$ ， b 长度为 1 的前缀 $\{2\}$ 是任意一个的子序列，故答案为 1。

当 $x = 4$ ，有两种不走重复点的路径 $\{1, 2, 4\}$ 和 $\{1, 3, 4\}$ ，对应的权值序列为 $\{3, 2, 4\}$ 和 $\{3, 1, 4\}$ ，不存在 b 的前缀满足同时是这两个的前缀，故答案为 0。

【样例 2】

见选手目录下的 `path2.in/path2.ans`。
该样例符合 *Subtask 2* 的限制。

【样例 3】

见选手目录下的 `path3.in/path3.ans`。
该样例符合 *Subtask 4* 的限制。

【数据范围】

对于 100% 的数据，有 $1 \leq n, t, s \leq 5 \times 10^5, 1 \leq m \leq 10^6, 1 \leq a_i, b_i \leq n$ 。
保证图联通。

子任务编号	$n \leq$	$m \leq$	$s \leq$	分值
1	10	20	n	15
2	3000	5000	n	10
3	10^5	$n - 1$	n	15
4	10^5	2×10^5	20	20
5	10^5	2×10^5	n	10
6	5×10^5	10^6	n	30

回环 (tree)

【题目背景】

小 X 在 Dongchen University 上学，可各种各样的选课、频繁地调整上课地点让她辗转于奔波中，她不得不规划一下自己的课程时间与位置……

【题目描述】

Dongchen University 有 n 个上课地点，有 $n - 1$ 条带权边将这 n 个上课地点连接起来，换言之，这 n 个点形成了一棵树。

小 X 一天有 k 门课程，对应 k 个地点，每天上课前她都可以任意乘车到某一上课地点，下车后按任意顺序依次步行到所有上课地点。

由于车在小 X 下车后没有移动，因此小 X 需要上完所有课后回到起始点。

小 X 是一个时间管理大师，她想知道最长的步行距离以至于规划自己的课程时间，因此她向你求助。

由于小 X 的上课位置可能存在变动，会按时间顺序依次存在 m 次将原来在 x 点上课的课程改到在 y 点上课（修改对后面都有影响，即不独立），你需要输出未修改和每次修改后的最长距离。

【输入格式】

从文件 `tree.in` 中读入数据。

第一行输入三个正整数 n, m, k ，含义如题述。

第 $2 \sim n$ 行每行三个整数 u, v, w ，表示 u, v 之间存在一条长度为 w 的边。

接下来一行 k 个整数，表示小 X 初始的上课地点。

接下来 m 行每行两个整数 x, y ，含义如题述。

【输出格式】

输出到文件 `tree.out` 中。

输出 $m + 1$ 行每行一个整数，表示修改前和 $i - 1$ 次修改后小 X 的最长步行距离。

【样例输入 1】

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

7	2	1	2	
8	1	2	5	6
9	5	4		
10	2	3		

【样例输出 1】

1	18
2	24
3	36

【样例解释 1】

对于没修改的情况可以采用 $1 \rightarrow 2 \rightarrow 6 \rightarrow 5$ 的方式上课, 距离为 $2+3+7+6 = 18$, 可以证明不存在更长的距离。

第一次修改把 5 改为 4, 可以采用 $1 \rightarrow 4 \rightarrow 6 \rightarrow 2$ 的方式上课, 距离为 $9 + 10 + 3 + 2 = 24$, 可以证明不存在更长的距离。

第二次修改把 2 改为 3, 可以采用 $1 \rightarrow 6 \rightarrow 3 \rightarrow 4$ 的方式上课, 距离为 $5 + 9 + 13 + 9 = 36$, 可以证明不存在更长的距离。

【样例 2】

见选手目录下的 *tree2.in/tree2.ans*。
该样例符合 Subtask 2 的限制。

【样例 3】

见选手目录下的 *tree3.in/tree3.ans*。
该样例符合 Subtask 4 的限制。

【样例 4】

见选手目录下的 *tree4.in/tree4.ans*。
该样例符合 Subtask 5 的限制。

【样例 5】

见选手目录下的 *tree5.in/tree5.ans*。
该样例符合 Subtask 7 的限制。

【数据范围】

对于 100% 的数据，有 $1 \leq n, m \leq 10^5, 1 \leq u, v, x, y \leq n, 1 \leq w \leq 10^7, 2 \leq k \leq n$ 。

保证 k 为偶数。

子任务编号	$n, m \leq$	特殊性质	分值
1	10	无	15
2	100	无	10
3	5000	无	10
4	7×10^4	A	15
5	7×10^4	B	10
6	7×10^4	无	15
7	10^5	无	25

特殊性质 A：满足 $\lfloor \frac{i}{2} \rfloor$ 与 $i (i \geq 2)$ 连边。

特殊性质 B： $k \leq \min\{n, 50\}$ 。

发绳 (rope)

【题目背景】

小 Z 是个爱美的女孩子，有一天收到了一根来自神秘人的发绳……

【题目描述】

小 Z 收到了一根长度为 L 的发绳，但小 Z 希望能将发绳裁成 n 段，并且只需要长度 $> K$ 的发绳做进一步的剪裁。

可小 Z 对数学一窍不通，只好每次在绳的任意位置（不一定为整数）裁剪一刀，经过 $n - 1$ 次后发绳就会变为 n 段。

她只需要长度 $> K$ 的发绳，她想知道这样经过 $n - 1$ 次裁剪后，可用绳段数为 i 的概率。

由于她不知道能得到多少有用的绳段，你需要对 $\forall i \in [0, n]$ 都输出答案，对 998244353 取模。

【输入格式】

从文件 `rope.in` 中读入数据。

共一行三个整数 n, L, K ，表示发绳裁剪后的段数、绳初始的长度和需要的小 Z 需要的最短长度。

【输出格式】

输出到文件 `rope.out` 中。

输出一行 $n + 1$ 个整数，第 i 个整数表示可用段数为 $i - 1$ 的概率对 998244353 取模后的结果。

【样例输入 1】

1275

【样例输出 1】

1570425345 427819009 0

【样例解释 1】

将长度为 7 的发绳随机裁剪 1 次，若裁剪位置在 $[2, 5]$ ，则不存在一条可用绳段，概率为 $\frac{3}{7} \equiv 570425345 \pmod{998244353}$ 。

若裁剪位置在 $(0, 2) \cup (5, 7)$ ，则存在一条可用绳段，概率为 $\frac{4}{7} \equiv 427819009 \pmod{998244353}$ 。

不存在裁剪成 2 条可用绳段的方式，故概率为 0。

【样例 2】

见选手目录下的 *rope2.in/rope2.ans*。

该样例符合 *Subtask 2* 的性质。

【样例 3】

见选手目录下的 *rope3.in/rope3.ans*。

该样例符合 *Subtask 3* 的性质。

【样例 4】

见选手目录下的 *rope4.in/rope4.ans*。

该样例符合 *Subtask 5* 的性质。

【样例 5】

见选手目录下的 *rope5.in/rope5.ans*。

该样例符合 *Subtask 8* 的性质。

【数据范围】

对于 100% 的数据，有 $1 \leq n \leq 5000, 1 \leq L \leq 10^6, 1 \leq K \leq L$ 。

子任务编号	$n \leq$	$L \leq$	分值
1	2	10^6	5
2	3	10^6	5
3	8	8	10
4	8	10^6	5
5	100	100	15
6	100	10^6	10
7	300	300	15
8	300	10^6	5
9	5000	5000	10
10	5000	10^6	20