

2026 河南省实验中学模拟赛

CSP-S 模拟赛

时间：2026 年 9 月 27 日 08:00 ~ 13:00

题目名称	接龙游戏	彩虹魔法	复制粘贴	栈
题目类型	传统型	传统型	传统型	传统型
目录	chain	rainbow	copy	stack
可执行文件名	chain	rainbow	copy	stack
输入文件名	chain.in	rainbow.in	copy.in	stack.in
输出文件名	chain.out	rainbow.out	copy.out	stack.out
每个测试点时限	1.0 秒	1.0 秒	3.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	3	5	6	5
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	chain.cpp	rainbow.cpp	copy.cpp	stack.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. `main` 函数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. 评测在 HS 的机子上进行，其他我不知道
9. 题目真的很简单，大家放心 AK。

接龙游戏 (chain)

【题目描述】

给定 n 个字符串 $S_1, S_2, \dots, S_N$ 。每个 S_i ($1 \leq i \leq n$) 都是由小写英文字母组成的非空字符串，长度不超过 10，且互不相同。

Alice (先手) 和 Bob (后手) 进行接龙游戏。两人轮流操作，Alice 先手。每一回合，当前玩家需要选择一个整数 i ($1 \leq i \leq n$)，并满足以下两个条件：

- i 不能等于之前所有回合中已经被选择过的任意整数；
- 如果这不是第一回合，设上一次选择的整数为 j ，则 S_j 的最后一个字母必须等于 S_i 的第一个字母。

如果某位玩家无法选择满足条件的 i ，则该玩家输，另一位玩家获胜。

请判断当两人都采取最优策略时，谁会获胜。

【输入格式】

从文件 **chain.in** 中读入数据。

输入的第一行包含一个正整数 n ，表示字符串个数。

输入的第 $2 \sim n+1$ 行包含一个字符串，其中第 i 个字符串表示 S_i 。

【输出格式】

输出到文件 **chain.out** 中。

输出一行表示游戏的结果，如果先手 Alice 获胜，输出 **First**；如果后手 Bob 获胜，输出 **Second**。

【样例 1 输入】

```
1 6
2 enum
3 float
4 if
5 modint
6 takahashi
7 template
```

【样例 1 输出】

1 First

【样例 1 解释】

例如，游戏可以如下进行。注意，这个过程的两人的操作不一定是最优的。

- 先手 Alice 选择 $i = 3$ 。
- 后手 Bob 选择 $i = 2$ ，且 `if` 的最后一个字母和 `float` 的第一个字母相同。
- 先手 Alice 选择 $i = 5$ ，且 `float` 的最后一个字母和 `takahashi` 的第一个字母相同。
- 后手 Bob 无法选择 $i \neq 2, 3, 5$ 且 S_i 的第一个字母为 `i` 的字符串，因此失败。此时，先手 Alice 获胜。

【样例 2】

见选手目录下的 `chain/chain2.in` 与 `chain/chain2.ans`。
该组样例满足 $n \leq 10$ 。

【样例 3】

见选手目录下的 `chain/chain3.in` 与 `chain/chain3.ans`。
该组样例满足 $n \leq 16$ 。

【数据范围】

本题采用子任务捆绑测试，你只有通过这个子任务的所有测试点，才能获得这个子任务的分数。

对于所有数据，保证： $1 \leq n \leq 16$ ， S_i ($1 \leq i \leq n$) 是由小写英文字母组成的非空字符串，长度不超过 10，且 $S_i \neq S_j$ ($1 \leq i < j \leq n$)。

子任务	n	分值
1	≤ 4	10
2	≤ 10	20
3	≤ 16	70

彩虹魔法 (rainbow)

【题目背景】

一天, Anis 和她的助手 Yufiria 又开始研究新的魔导器了! 这次她们创造的工具可以释放一种散发着绚烂多彩魔法。

Anis 大喊: “Yufi, 你看! 这种魔法和彩虹一样美丽!”, “哇, 真是五彩缤纷!” Yufiria 惊讶地说。两人很快就沉浸到魔法的世界里……

但是, 新的问题不知不觉出现了。

【题目描述】

Anis 和 Yufiria 创造的魔法可以看做是一个树图, 即任意两点之间仅存在一条路径的无向图。她们正在看的每棵树都有这样的特点: 每条边上的颜色都是 k 种颜色中的一种。如果树上的一个路径是彩色的, 意味着这条路径上至少包含两种不同颜色的边。

不过随着时间的变化, 魔导器的魔力耗尽了, 但是 Anis 和 Yufiria 因为沉浸在二人世界中, 忘记记录试验数据了; 为了尽力挽回这次试验, 她们打算还原魔法的形态。Anis 和 Yufiria 还记得树上 m 条彩色路径的起点和终点。他们想知道: 满足条件的树的有多少种可能? 由于答案可能很大, 所以请将答案对 $10^9 + 7$ 取模。

【输入格式】

从文件 `rainbow.in` 中读入数据。

输入第一行包含三个整数 n, m, k , 分别表示树上的节点数、Anis 和 Yufiria 所记得的彩色路径的个数和树枝的颜色总数。

接下来 $n - 1$ 行, 第 $i + 1$ 行包含两个整数 a_i, b_i , 表示点 a_i 和点 b_i 之间有一条树边。

接下来 m 行, 第 $n + j$ 行包含两个整数 c_j, d_j , 表示从点 c_j 到点 d_j 之间有一条彩色路径。保证 c_j 和 d_j 不相邻。

【输出格式】

输出到文件 `rainbow.out` 中。

输出仅一行, 输出满足条件的树的可能数, 答案需对 $10^9 + 7$ 取模。

【样例 1 输入】

```
1 3 1 2
```

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

【样例 1 输出】

```
1 2
```

【样例 1 解释】

第一种情况是点 1 和点 2 之间的边涂颜色 1，点 2 和点 3 之间的边涂颜色 2。

第二种情况是点 1 和点 2 之间的边涂颜色 2，点 2 和点 3 之间的边涂颜色 1。

【样例 2 输入】

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

【样例 2 输出】

```
1 0
```

【样例 3 输入】

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

【样例 3 输出】

16

【样例 4】

见选手目录下的 *rainbow/rainbow4.in* 与 *rainbow/rainbow4.ans*。
该组样例满足子任务 2 的限制。

【样例 5】

见选手目录下的 *rainbow/rainbow5.in* 与 *rainbow/rainbow5.ans*。
该组样例满足子任务 4 的限制。

【样例 6】

见选手目录下的 *rainbow/rainbow6.in* 与 *rainbow/rainbow6.ans*。
该组样例满足子任务 5 的限制。

【数据范围】

本题采用子任务捆绑测试，你只有通过这个子任务的所有测试点，才能获得这个子任务的分数。

对于所有的数据，保证 $3 \leq n \leq 60, 1 \leq m \leq 15, 2 \leq k \leq 10^9, 1 \leq a_i, b_i, c_j, d_j \leq n$ 。

子任务	特殊性质	分值
1	$m = 1$	10
2	$m = 2$	15
3	每个树边最多属于 m 条彩色路径中的一条	10
4	$3 \leq n \leq 15, k = 2$	10
5	没有额外限制	55

复制粘贴 (copy)

LJZ 公司是一家以“没啥用发明”而闻名的公司。最近, LJZ 公司开发了一款名为“没啥用编辑器”的编辑器。

在这个编辑器中, 可以执行如下几种操作来输入某个字符串, 设 X 为屏幕上的字符串, Y 为剪切板中的字符串, 初始均为空串:

- 操作 A: 输入字符 c , 即将 X 更新为 $X + c$ 。
- 操作 B: 选择所有字符并剪切, 即将 Y 更新为 X , 并将 X 置为空串。
- 操作 C: 将剪切板中的字符串粘贴到当前字符串末尾, 即将 X 更新为 $X + Y$ 。

对于字符串或字符 x, y , $x + y$ 表示将 x 和 y 顺次拼接得到的结果。使用一次操作 A, B, C 分别要花费 A, B, C 单位时间。

你安装了“没啥用编辑器”, 并想要尽可能快地输入一个长度为 n 的字符串 S 。

你需要计算出最少需要花费多少时间。

【输入格式】

从文件 `copy.in` 中读入数据。

第一行一个整数 n 表示字符串长度。

第二行一个长度为 n 的字符串 S 表示要输入的字符串。

第三行一个整数 A 表示操作 A 的代价。

第四行一个整数 B 表示操作 B 的代价。

第五行一个整数 C 表示操作 C 的代价。

【输出格式】

输出到文件 `copy.out` 中。

一行一个整数表示输入字符串 S 最少要多少单位时间。

【样例 1 输入】

```
1 11
2 mississippi
3 10
4 5
5 2
```

【样例 1 输出】

```
1 88
```

【样例 1 解释】

以下是一组最优操作：

轮次	操作	解释	X	Y	代价	总时间
-	-	-	" "	" "	-	0
1	操作 A	输入字符	"s"	" "	10	10
2	操作 B	全选并剪切	" "	"s"	5	15
3	操作 C	在尾部粘贴	"s"	"s"	2	17
4	操作 C	在尾部粘贴	"ss"	"s"	2	19
5	操作 A	输入字符	"ssi"	"s"	10	29
6	操作 B	全选并剪切	" "	"ssi"	5	34
7	操作 A	输入字符	"m"	"ssi"	10	44
8	操作 A	输入字符	"mi"	"ssi"	10	54
9	操作 C	在尾部粘贴	"missi"	"ssi"	2	56
10	操作 C	在尾部粘贴	"mississi"	"ssi"	2	58
11	操作 A	输入字符	"mississip"	"ssi"	10	68
12	操作 A	输入字符	"mississipp"	"ssi"	10	78
13	操作 A	输入字符	"mississippi"	"ssi"	10	88

【样例 2 输入】

```
1 16
2 aaaaaaaaaaaaaaaaaa
3 1
4 1
5 1
```

【样例 2 输出】

```
1 9
```

【样例 2 解释】

一组最优策略如下：

轮次	操作	解释	X	Y	代价	总时间
-	-	-	" "	" "	-	0
1	操作 A	输入字符	"a"	" "	1	1
2	操作 A	输入字符	"aa"	" "	1	2
3	操作 A	输入字符	"aaa"	" "	1	3
4	操作 A	输入字符	"aaaa"	" "	1	4
5	操作 B	全选并剪切	" "	"aaaa"	1	5
6	操作 C	在尾部粘贴	"aaaa"	"aaaa"	1	6
7	操作 C	在尾部粘贴	"aaaaaaaa"	"aaaa"	1	7
8	操作 C	在尾部粘贴	"aaaaaaaaaaaa"	"aaaa"	1	8
9	操作 C	在尾部粘贴	"aaaaaaaaaaaaaaaa"	"aaaa"	1	9

【样例 3 输入】

```

1 18
2 aababbbbababbbaabbb
3 1000000000
4 100000
5 10000000

```

【样例 3 输出】

```

1 8060200000

```

【样例 4】

见选手目录下的 *copy/copy4.in* 与 *copy/copy4.ans*。
该组样例满足子任务 3 的限制。

【样例 5】

见选手目录下的 *copy/copy5.in* 与 *copy/copy5.ans*。
该组样例满足子任务 4 的限制。

【数据范围】

本题采用子任务捆绑测试，你只有通过这个子任务的所有测试点，才能获得这个子任务的分数。

对于所有的数据，保证 $1 \leq n \leq 2500$, S 是一个长度为 n 的小写字母串， $1 \leq A, B, C \leq 10^9$ 。

子任务	特殊性质	分值
1	$n = 3$	1
2	S 只包含字符 a	5
3	$n \leq 30$	14
4	$n \leq 200$	10
5	$n \leq 1000$	32
6	没有额外限制	38

栈 (stack)

【题目描述】

有 n 个栈，编号为 1 到 n 。还有 m 次操作，操作有三种：

1. 给定 l, r, x, y ，表示给编号在区间 $[l, r]$ 内的栈都压入 x 个 y 。
2. 给定 l, r, w ，表示让编号在区间 $[l, r]$ 内的栈执行弹栈操作 w 次，这里弹栈操作是指，如果栈为空，则什么都不做；否则弹出栈顶。
3. 给定 k, p, q ，表示查询编号为 k 的栈里从栈底开始数第 p 个到第 q 个元素之和，如果栈不存在第 i 个元素，则视为栈的第 i 个元素为 0。

【输入格式】

第一行两个整数 n, m 。

接下来 m 行每行描述了一次操作，形式如下：

- 1 $l\ r\ x\ y$ ：含义如上。
- 2 $l\ r\ w$ ：含义如上。
- 3 $k\ p\ q$ ：含义如上。

【输出格式】

对于每个询问，输出一个数表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 4
```

2	5
3	2
4	2

【样例 2】

见选手目录下的 `stack/stack2.in` 与 `stack/stack2.ans`。
该组样例满足子任务 2 的限制。

【样例 3】

见选手目录下的 `stack/stack3.in` 与 `stack/stack3.ans`。
该组样例满足子任务 3 的限制。

【样例 4】

见选手目录下的 `stack/stack4.in` 与 `stack/stack4.ans`。
该组样例满足子任务 4 的限制。

【样例 5】

见选手目录下的 `stack/stack5.in` 与 `stack/stack5.ans`。
该组样例满足子任务 5 的限制。

【子任务】

对于所有的数据，保证 $1 \leq n, m \leq 10^5, 1 \leq x, y \leq 10^5, 1 \leq w \leq 10^{10}, 1 \leq p \leq q \leq 10^{10}$ 。

子任务	特殊性质	分值
1	$n, m \leq 5000$	18
2	保证不存在 2 操作	21
3	对于所有操作 1，保证 $y = 1$	14
4	对于所有 3 操作，保证 $p = 1, q = 10^{10}$	10
5	无特殊限制	32