

CSP2023模拟赛

题目名称	机器人	优秀的拆分	卡西欧质数	推车
题目类型	传统型	传统型	传统型	传统型
可执行文件名	robot	excellent	casio	car
输入文件名	robot.in	excellent.in	casio.in	car.in
输出文件名	robot.out	excellent.out	casio.out	car.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限	1024 MiB	1024 MiB	1024 MiB	1024 MiB
子任务/测试点数目	4	20	20	3
是否等分	否	是	是	是

提交源文件程序名

对于C++语言	robot.cpp	excellent.cpp	casio.cpp	car.cpp
---------	-----------	---------------	-----------	---------

编译选项

对于C++语言	-lm -O2 -std=c++17
---------	--------------------

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()的返回类型必须是int，程序正常结束时返回值必须是0。
3. 选手提交的程序代码文件请在**个人目录下以及子文件夹内各放一份**。
4. 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）。
5. 选手提交的程序源文件必须不大于100KB。
6. 程序可使用的栈空间内存限制于题目的内存限制一直。
7. 使用std::deque等STL容器时，请注意其内存空间消耗。
8. 评测时采用的机器配置为 AMD Ryzen 7 5800H with Radeon Graphics，内存16GiB。上述时限以此配置为准。
9. 评测在Windows 10下进行，使用LemonLine进行评测。

机器人 (robot)

一棵 n 个节点的树上有 m 个速度很快的机器人。第 i 个机器人初始在节点 p_i 。初始没有两个机器人在同一个节点。

一位长者参观了这棵树，认为这些机器人太难看了。他觉得第 i 个机器人应该移动到 q_i 去。

你需要移动这些机器人。每秒钟你可以通过远程遥控发出一次指令，让一个机器人从当前节点移动到一个和当前节点相邻的节点。任意时刻，如果两个机器人位于同一个节点，它们就会同时爆炸，于是你的移动过程中不允许出现这种情况。

长者认为应该节能，于是要求机器人必须以 p_i, q_i 两点在树上的最短路径从 p_i 移动到 q_i 。你需要判断是否存在一组合法的移动方案。

【输入格式】

本题有多组测试数据。

第一行一个正整数 T ，表示数据组数。

对于每组数据，第一行两个正整数 n 。

接下来 $n - 1$ 行，每行两个正整数 u, v ，表示树中存在一条边 (u, v) 。

接下来一行一个正整数 m 。

接下来 m 行，每行两个正整数 p_i, q_i 。

【输出格式】

对于每组数据，如果村子啊合法的解，输出一行一个字符串 `Yes`，否则输出一行一个字符串 `No`。

【输入输出样例1】

robot.in	robot.out
2	
6	
1 2	
1 3	
2 4	
2 5	
2 6	
2	
2 1	No
6 3	Yes
5	
1 2	
2 3	
3 4	
4 5	
2	
1 3	
2 5	

【输入输出样例1说明】

第一组数据中，不存在任何一种合法的移动方案。

考虑以下方案：让第一个机器人从节点 2 移动到节点 5；然后，让第二个机器人从节点 2 开始，依次移动到节点 6, 1, 3；最后让第一个机器人从节点 5 开始，依次移动到节点 2, 1。虽然过程中没有机器人发生碰撞，但是第一个机器人没有按照最短路径移动，因此它不是一种合法方案。

第二组数据中，以下移动方案是合法的：让第二个机器人从节点 2 开始，依次移动到节点 3, 4；然后，让第一个机器人从节点 1 开始，依次移动到 2, 3；最后，让第二个机器人从节点 4 移动到节点 5。

【输入输出样例2】

robot.in	robot.out
2	
9	
1 2	
1 3	
2 4	
2 5	
5 6	
5 7	
7 8	
8 9	
3	Yes
5 6	No
4 2	
9 3	
4	
1 2	
2 3	
3 4	
2	
1 3	
4 2	

【输入输出样例2说明】

第一组数据中，按照以下顺序移动：

1. 第一个机器人从节点 5 移动到节点 6；
2. 第三个机器人从节点 9 依次移动到 8, 7, 5, 2, 1。
3. 第二个机器人从节点 4 移动到节点 2。

【数据规模与约定】

本题采用捆绑测试。

对于所有测试数据，满足

$1 \leq T \leq 1000, 1 \leq m \leq n \leq 1.2 \times 10^5, \sum n \leq 1.2 \times 10^5, 1 \leq p_i, q_i, u_i, v_i \leq n, p_i$ 互不相同, q_i 互不相同, 对于同一个 i , 保证 $p_i \neq q_i$ 。

- Subtask 1(40 pts) : $T \leq 100, n \leq 250, m \leq 100$.
- Subtask 2(15 pts) : $T \leq 20, m \leq 500$.
- Subtask 3(5 pts) : $\forall i \in [1, n), u_i = i, v_i = i + 1$.
- Subtask 4(40 pts) : 无特殊限制

优秀的拆分 (cexcellent)

【题目描述】

如果一个字符串可以被拆分为 $AABB$ 的形式，其中 A 和 B 是任意非空字符串，则我们称该字符串的这种拆分是优秀的。

例如，对于字符串 `aabaabaa`，如果令 $A = \text{aab}$ ， $B = \text{a}$ ，我们就找到了这个字符串拆分成 $AABB$ 的一种方式。

一个字符串可能没有优秀的拆分，也可能存在不止一种优秀的拆分。

比如我们令 $A = \text{a}$ ， $B = \text{baa}$ ，也可以用 $AABB$ 表示出上述字符串；但是，字符串 `abaabaa` 就没有优秀的拆分。

现在给出一个长度为 n 的字符串 S ，我们需要求出，在它所有子串的所有拆分方式中，优秀拆分的总个数。这里的子串是指字符串中连续的一段。

以下事项需要注意：

- 出现在不同位置的相同子串，我们认为不同的子串，它们的优秀拆分均会被记入答案。
- 在一个拆分中，允许出现 $A = B$ 。例如 `cccc` 存在拆分 $A = B = \text{c}$ 。
- 字符串本身也是它的一个子串。

宇宙射线击中了数据，现在字符串中会有一部分位置为 `?`，定义两个字符串相等，当且仅当其长度相等，且每一组对应位均相等或至少有一个 `?`。比如 `a?c` 和 `abc` 相等，但 `a?c` 和 `??b` 不相等。某组拆分 (A, B) 是字符串 T 的优秀拆分当且仅当 A, B 均为由小写字母组成的非空字符串，且 $T = AABB$ 。对于同一字符串，定义两种拆分方式不同当且仅当其中 A 或 B 的长度不同。

【输入格式】

本题有多组测试数据。

第一行只有一个整数 T ，表示数据的组数。

接下来 T 行，每行包含一个仅由 `?` 和英文小写字母构成的字符串 S ，意义如题所述。

【输出格式】

输出 T 行，每行包含一个整数，表示字符串 S 所有子串的所有拆分中，总共有多少个是优秀的拆分。

【输入输出样例1】

excellent.in	excellent.out
4	3
aabbbb	5
ccccc	4
aabaabaabaa	7
bbaabaababaaba	

【输入输出样例 1 说明】

由于 A, B 要求非空, 因此长度 ≤ 3 的子串均无优秀的拆分。 `????` 有且仅有一组优秀的拆分即 $|A| = |B| = 1$ 。尽管根据 `?` 处填入字符不同, A 和 B 分别填入任意一个小写字母均为合法方案, 但由于 $|A|, |B|$ 均相等, 故视为一种。

【输入输出样例2】

excellent.in	excellent.out
2	
????	1
?ab??	1

【输入输出样例 1 说明】

第二组数据中, 仅有子串 `?ab?` 有优秀的拆分 $A = a, B = b$ 。

【数据规模与约定】

以下的 n 均指 T 的长度。

- 对于前 20% 的测试数据, $n \leq 50$ 。
- 对于前 40% 的测试数据, $n \leq 200$ 。
- 另有 10% 的测试数据, S 中不含 `?`。
- 另有 15% 的测试数据, S 中所有字符均为 `?`。
- 另有 20% 的测试数据, $1 \leq n \leq 2000$ 。

对于所有的测试数据, $1 \leq T \leq 10, 1 \leq n \leq 3000, S$ 仅包含 `?` 和英文小写字母。

卡西欧质数 (casio)

【题目描述】

报数游戏实在太无聊了。

于是小 A 发明了一种新的报数游戏：从小到大，每个人依次报质数。受技术限制，他们只能通过卡西欧计算器验证是否是质数，而计算器无法验证大于某个特定值的质因子（即若一个数的质因子均大于某个特定值，则会被判定为质数）。无论如何，小 A 决定以卡西欧作为质数判断依据，并定义为“卡西欧质数”。

简要题面：

给定一个质数 P ，定义一个数 x 为“卡西欧质数”当且仅当 x 为质数或 x 的最小素因子严格大于 P 。

T 组询问，每次给定一个 n ，求不小于 n 的最小“卡西欧质数”。

【输入格式】

第一行两个正整数 T, P 。

之后 T 行，每行一个正整数 n ，表示一组询问。

【输出格式】

对于每组询问，输出一行，一个数，表示不小于 n 的最大“卡西欧质数”。

【输入输出样例1】

casio.in	casio.out
2 3	23
22	25
25	

【数据规模与约定】

- 对于前 20% 的测试数据，满足 $n \leq 1000$ 。
- 对于前 60% 的测试数据，满足 $n \leq 10^6$ 。
- 另有 20% 的测试数据，满足 $P = 998244353$ 。

对于 100% 的测试数据，满足 $1 \leq T \leq 3 \times 10^5, 1 \leq P \leq 10^9, 2 \leq n \leq 3 \times 10^7$ ，保证 P 为质数。

推车 (car)

【题目描述】

假期到了！小 M，小 N 和小 S 决定推着橡树车，从一条数轴上的 s 点开始，去 H 市——位于数轴上 s 右侧的 t 点找她们的好朋友玩。为了让大家得到充分的休息，她们决定轮流让三个人中的某一个来推车。

橡树车上能够存储 m 个单位的零食。小 M，小 N 和小 S 喜欢的是三种不同的零食，分别记作 SM，SN 和 SS。出发前，橡树车上因为小 M 的个人喜好，装满了 m 个单位的 SM。她们中的每一个人推每一个单位长度，就会把橡树车上自己喜欢的零食吃掉一个单位。如果车上没有自己喜欢的零食了，那么当前推车的人就会立刻停止推车。

路上有 n 个零食店，每个零食店只出售 SM，SN 和 SS 中的某一种零食。第 i 个零食店位于数轴上的 t_i ，出售的零食种类是 k_i 。同一个点上可能有多个零食店。零食店的零食储量可以看做无穷大，因此她们可以在不超过橡树车存储量的前提下，在零食店购买任意数量的零食。

因为小 S 的地位高于小 N，小 N 的地位高于小 M，因此，当多个人能够推车时，一定是小 M 优先推车，然后是小 N，最后是小 S；另外，她们希望小 S 推车的长度尽可能小，在此前提下，小 N 推车的长度尽可能小。

现在她们要去 H 市 q 次，每次的起点可能不同，即：第 i 次从 s_i 点出发，去往 s_i 右侧的 t 点。你需要求出，每次行程小 S 推车的长度，和满足小 S 推车的长度尽可能小的前提下，小 N 推车的长度；或者指出，她们不可能到达 H 市。

【输入格式】

第一行四个整数 m, t, n, q 。

接下来 n 行，每行一行一个整数 t_i 和一个字符串 k_i 。

接下来一行 q 个整数 s_i 。

【输出格式】

输出 q 行，第 i 行两个整数，分别表示第 i 次行程小 S 推车的长度，和满足小 S 推车的长度尽可能小的前提下，小 N 推车的长度。如果本次行程无法到达 H 市，输出一行一个整数 -1 。

【输入输出样例1】

car.in	car.out
6 18 4 4	
2 SN	
6 SS	8 1
4 SN	8 0
12 SS	2 0
3	8 2
4	
10	
2	

【输入输出样例1说明】

以起点为 3 时为例：小 M 推车到 4，并补充 1 单位的 SN；然后推车到 6，补充 2 单位的 SS；然后推车到 9，换小 N 推车到 10；然后，换小 S 推车到 12，补充 6 单位的 SS，小 S 推车到 18，到达 H 市。

【数据规模与约定】

本题采用捆版测试。

- Subtask 1 (20 pts) : $1 \leq n, q \leq 100$ 。
- Subtask 2 (20 pts) : $1 \leq n, q \leq 5000$ 。
- Subtask 3 (60 pts) : 无特殊限制。

对于全部数据，满足

$1 \leq m, t \leq 10^9, -10^9 \leq t_i \leq 10^9, -10^9 \leq s_i \leq t, 1 \leq n, q \leq 2 \times 10^5, k_i \in \{\text{SM}, \text{SN}, \text{SS}\}$ 。