

А. Динамический Лес

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод
вывод: стандартный вывод

Вам нужно научиться обрабатывать 3 типа запросов:

1. Добавить ребро в граф («link»).
2. Удалить ребро из графа («cut»).
3. По двум вершинам a и b вернуть длину пути между ними (или -1 , если они лежат в разных компонентах связности) («get»).

Изначально граф пустой (содержит N вершин, не содержит ребер). Гарантируется, что в любой момент времени граф является лесом. При добавлении ребра гарантируется, что его сейчас в графе нет. При удалении ребра гарантируется, что оно уже добавлено.

Входные данные

Числа N и M ($1 \leq N \leq 10^5 + 1$, $1 \leq M \leq 10^5$) — количество вершин в дереве и, соответственно, запросов.

Далее M строк, в каждой строке команда («link» или «cut», или «get») и 2 числа от 1 до N — номера вершин в запросе.

Выходные данные

В выходной файл для каждого запроса «get» выведите одно число — расстояние между вершинами, или -1 , если они лежат в разных компонентах связности.

Примеры

входные данные

```
3 7
get 1 2
link 1 2
get 1 2
cut 1 2
get 1 2
link 1 2
get 1 2
```

выходные данные

```
-1
1
-1
1
```

входные данные

```
5 10
link 1 2
link 2 3
link 4 3
cut 3 4
get 1 2
get 1 3
get 1 4
get 2 3
get 2 4
get 3 4
```

выходные данные

```
1
2
-1
1
-1
-1
```

В. Асхат и дерево

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Это интерактивная задача.

Дано двоичное дерево поиска размера n , в каждой вершине есть значение от 1 до n . Петух по имени Асхат каждый день нумерует вершины числами от 1 до n , даёт вам номер корня r и просит вас найти номер вершины со значением x .

Вы можете совершать запросы — по номеру вершины узнать значение в ней и номера её детей. Пусть максимальное расстояние от корня до вершины в i -й день равно d . Тогда в i -й день вы можете совершить не более, чем d запросов. За все дни вы можете совершить не более, чем 70000 запросов.

Также вы можете менять детей у каждой вершины. Пусть в i -й день длина пути от корня до вершины с номером X равна S . Тогда вам в i -й день разрешено сделать не более, чем S запросов вида «установить у вершины новых детей». За все дни вы можете совершить не более, чем 70000 запросов этого типа.

Протокол взаимодействия

В первой строке ввода даны числа n и q ($1 \leq n \leq 2000$, $1 \leq q \leq 2000$) — размер дерева и число запросов, соответственно.

Для каждого запроса даны числа r и x ($1 \leq r, x \leq n$) — номер корня дерева и значение, вершину с которым требуется найти. Вы не сможете считать эти числа для следующего запроса, пока не дадите ответ на текущий.

Чтобы обратиться к вершине с номером i выведите «`val i`» в отдельной строке. В ответ даются три числа val , L и R ($1 \leq val \leq n$, $0 \leq L, R \leq n$) — значение в этой вершине и номера левого и правого ребёнка, соответственно. В случае, если у вершины нет левого или правого ребёнка, $L = 0$ или $R = 0$, соответственно.

Чтобы поменять детей вершины с номером i на вершины с номером L и R выведите «`change i L R`» в отдельной строке. Чтобы у вершины с номером i не было левого или правого ребёнка, выведите 0 вместо L или R , соответственно. После выполнения этого запроса граф может перестать быть двоичным деревом поиска.

Если искомое значение находится в вершине с номером i и вы совершили все нужные изменения, выведите «`confirm i`» в отдельной строке. После этого вершины перенумеруются, а на ввод будет дан новый запрос. Если на момент выполнения этого запроса граф не является двоичным деревом поиска, вы получите вердикт «Wrong answer».