Zurikela's Graph

by nikasvanidze

Problem | Submissions | Leaderboard | Discussions | Editorial

Zurikela is creating a graph with a special graph maker. At the begining, it is empty and has no nodes or edges. He can perform 3 types of operations:

1. $A\ x$: Create a set of x new nodes and name it $set\text{-}K$.
2. $B\ x\ y$: Create edges between nodes of $set\text{-}x$ and $set\text{-}y$.
3. $C\ x$: Create a set composed of nodes from $set\text{-}x$ and its directly and indirectly connected nodes, called $set\text{-}K$. Note that each node can only exist in one set, so other sets become empty.

The first set's name will be $set\text{-}1$. In first and third operation K is referring to the index of new set:

```
K = [index of last created set] + 1
```

Create the graph by completing the Q operations specified during input. Then calculate the maximum number of independent nodes (i.e.:how many nodes in the final graph which don't have direct edge between them).

Input Format

The first line contains Q.
The Q subsequent lines each contain an operation to be performed.

Constraints
$1 \leq Q \leq 10^5$.
For the first operation, $1 \leq x \leq 10^4$.
For the second operation, $x < y$ and all ys are *distinct*.
For the second and third operation, it's guaranteed that $set\text{-}x$ and $set\text{-}y$ exist.

Output Format

Print maximum number of *independent nodes* in the final graph (i.e.: nodes which have no direct connection to one another).

Sample Input

```
8
A 1
A 2
B 1 2
C 1
A 2
A 3
B 3 4
B 4 5
```

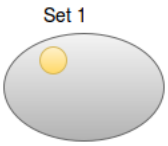
Sample Output

```
5
```

Explanation

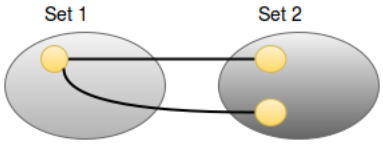
There are 8 operations.

After first operation $(A\ 1)$:

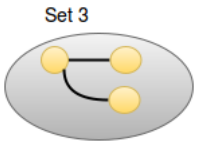


After second operation $(A\ 2)$:

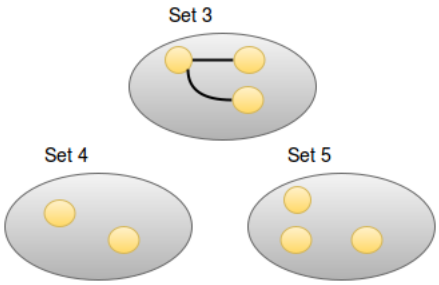
After third operation $(B\ 1\ 2)$:



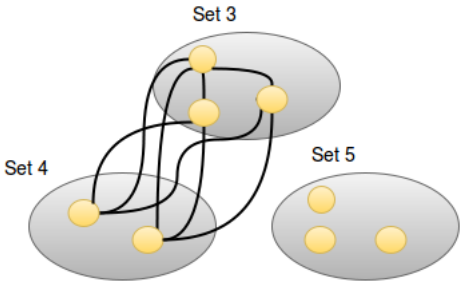
After fourth operation $(C\ 1)$:



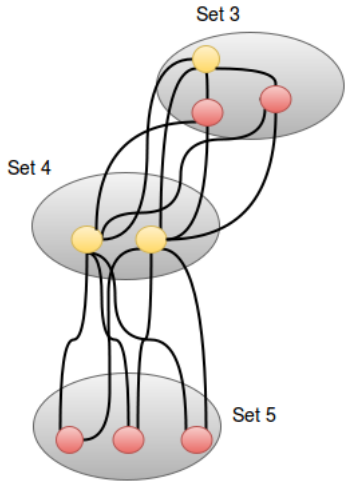
After fifth and sixth operation $(A\ 2)$ and $(A\ 3)$:



After seventh operation $(B\ 3\ 4)$:



After eigth operation $(B\ 4\ 5)$:



There are 2 independent nodes in $set\text{-}3$ and 3 independent nodes in $set\text{-}5$, so we print their sum (5) as our answer.

Submissions:73

Max Score:80

Difficulty: Hard

Rate This Challenge:

More

Current Buffer (saved locally, editable) Java 7

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  public class Solution {
8
9      public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11     }
12 }
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code