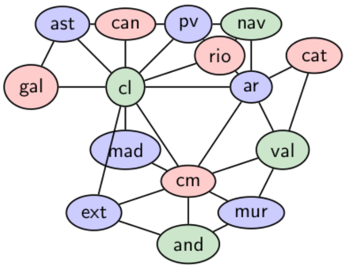
Jutge.org

The Virtual Learning Environment for Computer Programming

Graph Coloring X33462_en

Write a program in Python that, using the **optilog** library, finds a coloring for a given graph.



In order to use the optilog library, the program has to include something like:

```
from optilog.solvers.sat import *
...
solver = Glucose41()
solver.add_clauses(...)
solver.solve()
solver.model()
```

Input

The input is a text (in the stdin) with pairs of connected nodes. For instance, the text:

```
a b
a c
b c
b d
c d
```

Output

The output is also a text (in the stdout) where in every line there is a list of nodes with the same color. In this example:

```
{a, d}
{b}
{c}
```

Notice that the order of the lines and the order inside each line is not relevant. In this example, there are three lines because this is the minimum number of required colors and node 'a' and 'd' can get the same color because they are not connected.

Sample input 1

```
a b
a c
b c
b d
c d
```

Sample output 1

```
{c}
{d, a}
{b}
```

Sample input 2

```
ast gal
ast can
ast cl
gal cl
can cl
can pv
pv cl
pv rio
pv nav
rio cl
rio ar
rio nav
nav ar
cat ar
cat val
ar cl
ar val
ar cm
cl mad
cl ext
cl cm
cm mad
cm ext
cm and
cm mur
cm val
val mur
ext and
and mur
```

Sample output 2

```
{nav, and, val, cl}
{can, cm, rio, cat, gal}
{pv, ast, ext, mur, mad, ar}
```

Scoring

Samples have been selected in order to ensure that there exist a unique solution up to colors permutations.

Problem information

Author : Jordi Levy
Generation : 2023-11-30 16:25:25