# Count the Faces

Uva10178

Time: 3 seconds

Problem Description

- A planar graph is defined as follows Definition: A planar graph is one that can be drawn on a plane in such a way that there are **no "edge crossings,"** i.e. edges intersects only at their common vertices.

- The figure on the right shows a planar graph. The **six different faces** of the graph are colored with different colors and are also numbered from 1 to 6.

- You will have to **count the number of faces** of a given planar graph.
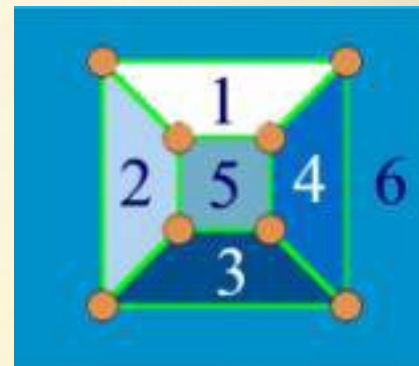
Input

- **The input contains several sets of inputs. Each set of input contains two integers N, E in the first line,**
  - **N denotes the number of nodes of the graph**
  - **E denotes the number of edges**
- **The next E lines contain the description of E edges of a planar graph. Each edge description contains two case sensitive English alphabets n_1 and n_2, which indicates that vertex n_1, and n_2 are connected by an edge.**
- **Input is terminated by end of file.**

Output

- For each set of input <u>print the number of faces</u> in that graph in a single line.

Sample Input / Output

N E   **number of nodes and edges**

1 0

3 3

A B

B C   **connected edges**

A C

1   **number of faces**

2

Sample Input / Output

N E

1 0

3 3
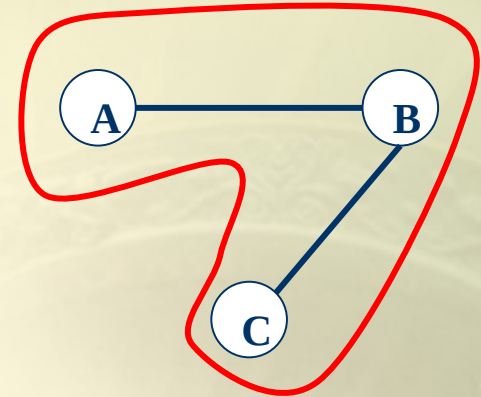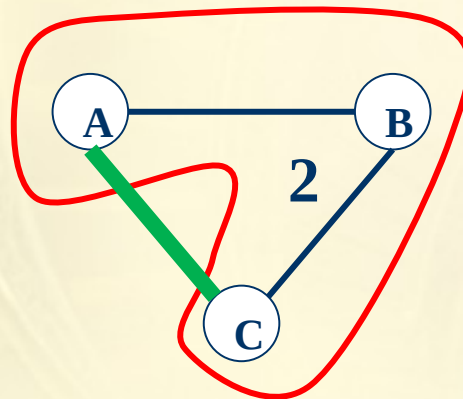
A B

B C

A C
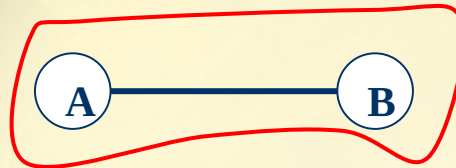
1

2

Disjoint Set

- **Operations**
  - *MakeSet*(x)

    x.parent := x
  - *Find*(x)
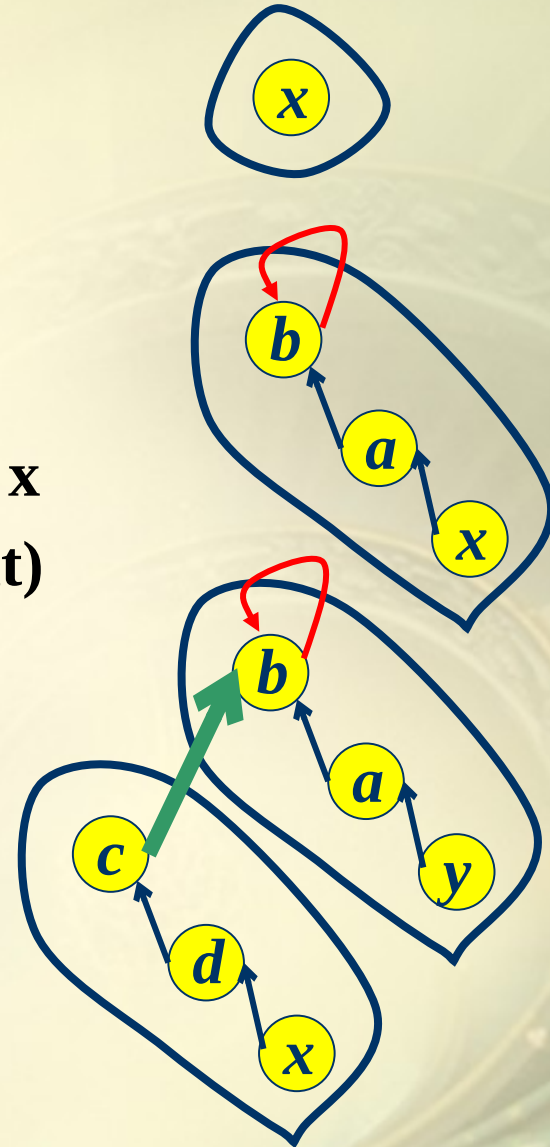
    if (x.parent == x) return x

    else return *Find*(x.parent)
  - *Union*(x,y)

    xRoot := *Find*(x)

    yRoot := *Find*(y)

    xRoot.parent := yRoot

Implement Disjoint Set

Operations

MakeSet(x)

    x.parent := x

Find(x)

    if (x.parent == x) return x

    else return *Find*(x.parent)

Union(x,y)

    xRoot := *Find*(x)

    yRoot := *Find*(y)

    xRoot.parent := yRoot

x	0	1	2	3	4	5
x.parent	0	1	2	3	4	5

x	0	1	2	3	4	5
x.parent	0	2	5	3	5	5

Find(1)=5

x	0	1	2	3	4	5
x.parent	0	2	5	3	5	5

Union(4,5)
Union(1,2)
Union(1,4)

Implement Disjoint Set
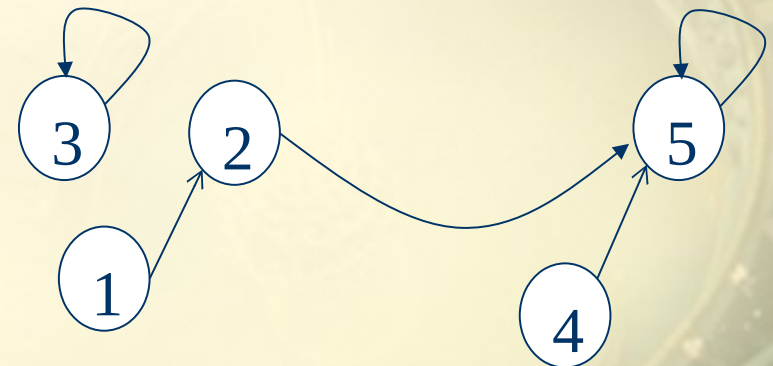
Count the faces

⬦**For each input A connects B**
  if Find(A)≠Find(B)
    Union(A,B)
  else
    face count ++