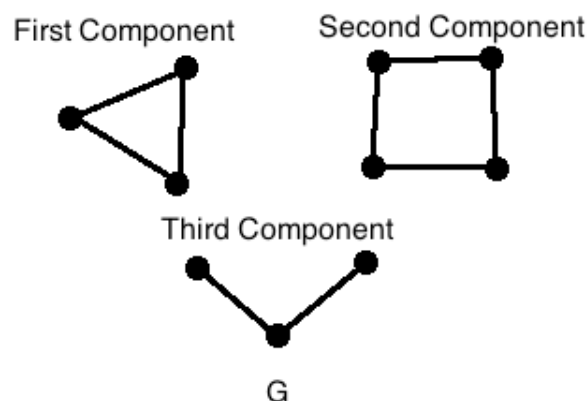


Component

หลังจากทราบวิธีท่องกราฟแล้ว มาดูกันว่าสามารถเอามาประยุกต์ใช้ทำอะไรได้บ้าง

สังเกตว่ากราฟ ๆ หนึ่งไม่จำเป็นจะต้องเชื่อมต่อกันหมดก็ได้ อาจจะอยู่แยกกันเป็นเกาะ ๆ โดยเกาะหนึ่งจะเรียกว่าเป็น **Connected Component** (หรือ **Component** เฉย ๆ) ของกราฟ

ยกตัวอย่าง กราฟดังภาพมี 3 component



ในการเขียนโปรแกรมนับ component สามารถทำได้โดยเริ่ม dfs จากโหนด 1 เพื่อหาว่ามีโหนดไหนที่สามารถไปมาหากันจากโหนด 1 ได้ โหนดที่เจอจะนับเป็น 1 component

หลังจาก DFS/BFS เสร็จแล้ว หากพบว่ายังมีโหนดที่ยังไม่ได้ไป ก็ให้ลองไปเริ่ม DFS/BFS จากโหนดนั้นต่อ เพื่อหาว่า component ที่ติดกับโหนดนั้นมีโหนดใดบ้าง นับเป็นอีก 1 component ทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะท่องไปครบทุกโหนด

จงเขียนโปรแกรมที่นับจำนวน component ในกราฟไม่ระบุทิศทางที่กำหนดให้

Input

บรรทัดแรก ประกอบด้วยจำนวนเต็ม n และ m แทนจำนวนโหนด และจำนวนเส้นเชื่อมในกราฟ ($1 \leq n \leq 10^5, 0 \leq m \leq 10^5$)

บรรทัดที่ $1 + i$ ($1 \leq i \leq m$) ประกอบด้วยจำนวนเต็ม x_i และ y_i แสดงว่า มีเส้นเชื่อมระหว่างโหนดที่ x_i และ โหนดที่ y_i ($1 \leq x_i, y_i \leq n$)

รับประกันว่ากราฟที่กำหนดให้จะเป็นกราฟอย่างง่าย (Simple Graph)

Output

ตอบจำนวน component เท่านั้น

Example

Input

```
10 9
1 2
2 3
3 1
4 5
5 6
6 7
7 4
8 9
9 10
```

Output

```
3
```