

Depth-first Search

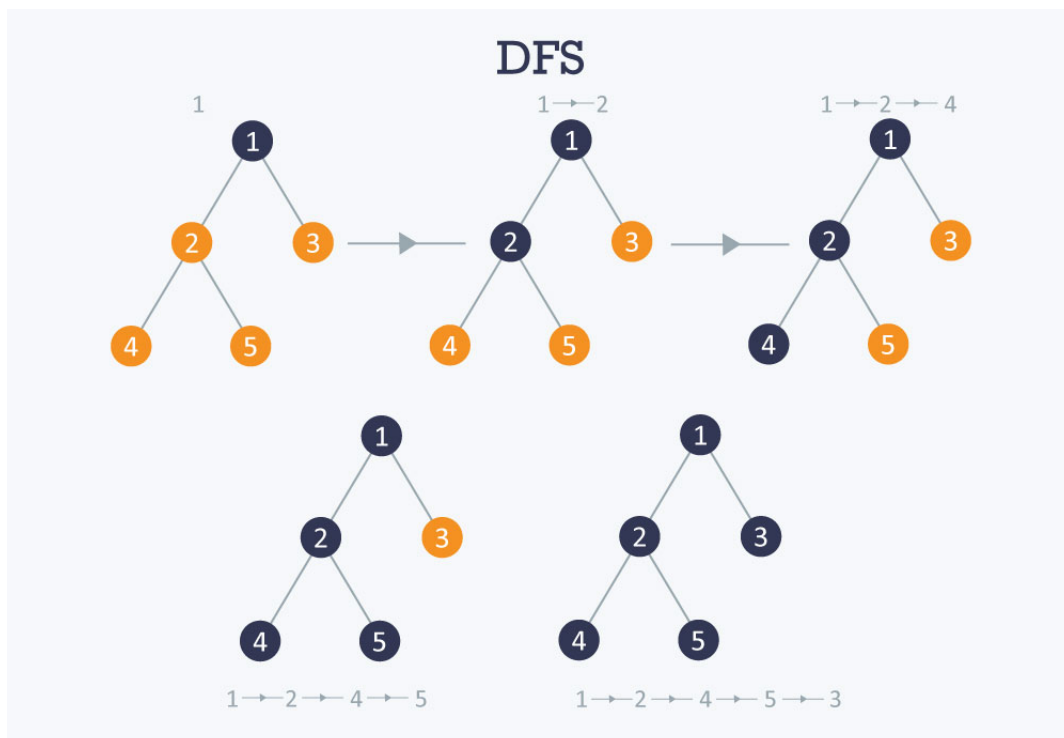
ในเมื่อเก็บกราฟได้แล้ว ต่อมา เราต้องเอากราฟมาใช้ประโยชน์
หนึ่งในวิธีที่ใช้ในการหาข้อมูลเกี่ยวกับกราฟ คือ **Graph Traversal** (การท่องเข้าไปในกราฟ) ซึ่ง
สามารถทำได้สองแบบ

- **Depth-first Search (DFS)** เน้นท่องเข้าไปให้สุดก่อน แล้วค่อยย้อนกลับขึ้นมา
- **Breadth-first Search (BFS)** ท่องเข้าไปตามลำดับความลึก เป็นชั้น ๆ ไป

ในที่นี้จะพูดถึง DFS ก่อน (กราฟในที่นี้เป็นกราฟแบบไม่มีทิศทาง) โดยการ DFS บนโหนด u จะมีขั้นตอนดังนี้

- 1) ปรีนที่โหนด u และจดว่าเคยท่องไปยังโหนด u แล้ว
- 2) สำหรับโหนดที่อยู่ติดกับ u แต่ละโหนด และยังไม่เคยท่องไป ให้ DFS บนโหนด v (recursive)

ยกตัวอย่าง การ DFS โดยเริ่มจากโหนด 1 ดังกราฟในภาพ (โหนดสีส้มคือโหนดที่ยังไม่เคยไป โหนดสีน้ำเงินคือโหนดที่เคยไปแล้ว)



ลำดับการทำงานของ dfs(1)

- print(1)
- dfs(2)
 - print(2)
 - dfs(4)
 - * print(4)
 - dfs(5)
 - * print(5)
- dfs(3)
 - print(3)

สรุปแล้ว จะได้ลำดับโหนดที่พิมพ์เป็น 1, 2, 4, 5, 3 นั่นเอง

อนึ่ง ต้องระวังว่าหากกราฟไม่ได้เป็นกราฟต้นไม้ดังตัวอย่าง ต้องระวังไม่ให้ DFS วงกลับมาที่โหนดเดิม เพราะฉะนั้นตลอดการทำงานของ DFS เมื่อพิมพ์โหนดใดไปแล้ว จะต้องจดไว้ว่าโหนดนั้นพิมพ์ไปแล้ว และจะไม่พิจารณา DFS โหนดนั้นอีก

จงเขียน โปรแกรมที่แสดงผลลัพธ์ของ Depth-first Search บนกราฟไม่ระบุทิศทาง เมื่อกำหนดจุดเริ่มต้นให้

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 u_i และ v_i แสดงว่า มีเส้นเชื่อมระหว่างโหนดที่ u_i และ โหนดที่ v_i ($1 \leq u_i, v_i \leq n$)

บรรทัดที่ $m + 2$ ประกอบด้วยจำนวนเต็ม s แทนจุดเริ่มต้นของ DFS

รับประกันว่ากราฟที่กำหนดให้จะเป็นกราฟอย่างง่าย (Simple Graph)

Output

ในหนึ่งบรรทัด ให้พิมพ์ตัวเลขลำดับโหนดที่ได้จากการ DFS

ระหว่างที่ทำ DFS เมื่อพิจารณาโหนดที่อยู่ติดกันมากกว่า 1 โหนด ให้พิจารณาโหนดที่มีหมายเลขน้อยที่สุดก่อน

Example #1

Input

```
5 4
1 2
1 3
2 4
2 5
1
```

Output

```
1 2 4 5 3
```

Example #2

Input

```
5 6
1 2
1 3
1 4
1 5
2 4
4 5
2
```

Output

```
2 1 3 4 5
```