

Degrees

(1 sec, 512 MB)

ในโจทย์ข้อนี้ จะต้องรับข้อมูลเส้นเชื่อมของ [Undirected unweighted simple graph](#) ซึ่งจะมาในรูปแบบของ [Adjacency Matrix](#)

จากนั้นจึงแสดงผลเป็นตัวเลขจำนวน $k + 1$ ตัวเลขโดยตัวเลขลำดับที่ i จะหมายถึงจำนวนปมที่มี degree เป็น i เมื่อ k คือดีกรีสูงสุดในกราฟ โดย i มีค่าตั้งแต่ 0 ถึง k

นิยาม

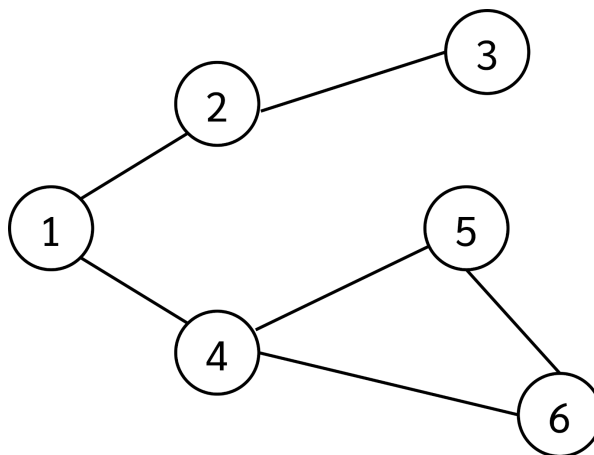
Undirected Unweighted Simple Graph

Undirected Unweighted Simple Graph เป็นกราฟชนิดหนึ่ง โดยกราฟ ประกอบด้วย ปม (node) และ เส้นเชื่อม (edge)

Undirected หมายถึง เส้นเชื่อมที่ไม่มีทิศทาง เช่น การมีเส้นเชื่อมจาก 1 ไป 2 แสดงว่า มีเส้นเชื่อมจาก 2 ไป 1 เช่นกัน

Unweighted หมายถึง เส้นเชื่อมไม่มีค่าน้ำหนัก

Simple Graph หมายถึง กราฟที่ไม่มีเส้นเชื่อมระหว่างปมมากกว่า 1 เส้น และไม่มีเส้นเชื่อมที่จุดเริ่มต้นและจุดสิ้นสุดเป็นปมเดียวกัน



รูปที่ 1: ตัวอย่าง Undirected Unweighted Simple Graph

Adjacency Matrix

(adjacent [adj.] ประชิด, อยู่ติดต่อกับ)

Adjacency Matrix คือ square matrix ที่บ่งบอกถึงกราฟ โดยแต่ละช่องใน matrix แสดงถึงว่ามีการเชื่อมต่อระหว่างปมสองปมหรือไม่ โดยค่า 1 หมายถึงมีการเชื่อมต่อ (edge) ระหว่างคู่ปมนั้นๆ ส่วน 0 หมายถึงไม่มีการเชื่อมต่อ

จากกราฟตัวอย่างในรูปที่ 1 สามารถเขียน Adjacency Matrix ได้ดังนี้

ปมที่	1	2	3	4	5	6
1	0	1	0	1	0	0
2	1	0	1	0	0	0
3	0	1	0	0	0	0
4	1	0	0	0	1	1
5	0	0	0	1	0	1
6	0	0	0	1	1	0

ข้อมูลนำเข้า

ข้อมูลจะมีหลายบรรทัดตามรูปแบบนี้

- บรรทัดแรกเป็นจำนวนเต็ม n แสดงถึงจำนวนปมในกราฟ
- จากนั้นอีก n บรรทัดจะประกอบด้วยจำนวนเต็ม n ตัว โดยแสดงถึง Adjacency Matrix ของกราฟ

ข้อมูลส่งออก

ประกอบด้วยหนึ่งบรรทัด โดยเป็น ตัวเลข $k+1$ ตัว โดยตัวเลขลำดับที่ i คือจำนวนปมที่มี degree เป็น i โดย k คือดีกรีสูงสุดในกราฟ

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
6 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0 1 1 0	0 1 4 1

ชุดข้อมูลทดสอบ

- 50% $n \leq 300$
- 50% $n \leq 5000$

หมายเหตุ

ข้อมูลนำเข้าและส่งออกของโปรแกรมนี้นี้มีเป็นจำนวนมาก กรณีใช้ภาษา C++ การทำงานตามปกติของ cin และ cout นั้นช้าเกินไป ขอให้เรียกคำสั่งดังต่อไปนี้เป็นคำสั่งแรกใน main function เพื่อเพิ่มความเร็วให้กับ cin และ cout

```
std::ios_base::sync_with_stdio(false); std::cin.tie(0)
```