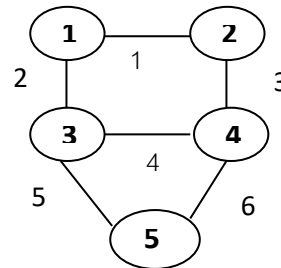


Graph: Weight Graph

ให้เขียนโปรแกรมรับข้อมูลกราฟแบบ Undirected Weighted Graph แล้วรับค่าเส้นทาง (path) ด้วยรายการโหนด (v_1, v_2, \dots, v_k) เพื่อตรวจสอบหาว่ากราฟมี path ที่เริ่มต้นจากโหนดต้นทาง (v_1) ไปยังโหนด v_i ต่างๆ ในรายการด้วยค่า path length สูงสุดเท่าใด โดยหากตรวจสอบว่าพบ cycle ใน path ให้หยุดการตรวจสอบเส้นทาง และแสดงผลรวมของ W ข้อมูลนำเข้าที่ทำการตรวจสอบหรือไม่

จากตัวอย่าง รูป Undirected Weighted Graph

ใช้สำหรับตัวอย่างที่ 1 ถึง 3



ตัวอย่างที่ 1 เส้นทางตรวจสอบคือ 4 2 1 3 5

พบว่า สามารถเดินทางตาม path ที่ให้ จากโหนดต้นทาง 4 ถึง โหนดปลายทาง 5 ด้วย path length = 4 และไม่พบ cycle ใน path ได้ผลรวมน้ำหนักเป็น $3 + 1 + 2 + 5 = 11$ จะแสดงผลลัพธ์ เป็น 4 11

ตัวอย่างที่ 2 เส้นทางตรวจสอบคือ 1 3 4 5 3 1

พบว่า สามารถเดินทางตาม path ที่ให้ จากโหนดต้นทาง 1 ถึง โหนดปลายทาง 3 เนื่องจากพบ Cycle ที่โหนด 3 ได้ผลรวมน้ำหนักเป็น $2 + 4 + 6 + 5 = 17$ จะแสดงผลลัพธ์ เป็น 4 17

ตัวอย่างที่ 3 เส้นทางตรวจสอบคือ 1 3 5 2 1

พบว่า สามารถเดินทางตาม path ที่ให้ จากโหนดต้นทาง 1 ถึง โหนดปลายทาง 5 ด้วย path length = 2 และไม่พบ cycle ใน path ได้ผลรวมน้ำหนักเป็น $2 + 5 = 7$ จะแสดงผลลัพธ์ เป็น 2 7

ข้อมูลเข้า มี V+2 บรรทัด

บรรทัดแรก เป็น เลขจำนวนเต็ม V และ E แทนจำนวน Vertices และ Edges ของกราฟ ($1 \leq V \leq 20, 1 \leq E \leq 100$)

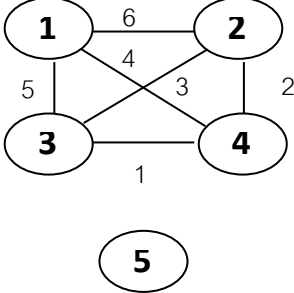
บรรทัดที่ 2 ถึง N+1 จะคู่ Vertices (u, v) แทนเส้นเชื่อมในกราฟ โดยที่ $1 \leq u_i, v_i \leq V$ และ $i=1 \dots E$

บรรทัดสุดท้าย เป็นเลขจำนวนเต็ม k+1 ค่า เป็นรายการโหนด (v_1, v_2, \dots, v_k) ของเส้นทาง (ค้นด้วยช่องว่าง โดยปิดท้ายด้วย -1 เป็นตัวกำหนดว่าไม่มีรายการโหนดแล้ว

ข้อมูลออก มี 1 บรรทัด

เป็นเลขจำนวนเต็ม 2 ค่า คือ path length และผลรวมน้ำหนักของ path ที่ได้

ตัวอย่าง

ข้อมูลเข้า	ข้อมูลออก
5 6 1 2 1 1 3 2 2 4 3 3 4 4 3 5 5 4 5 6 4 2 1 3 5 -1	4 11
5 6 1 2 1 1 3 2 2 4 3 3 4 4 3 5 5 4 5 6 1 3 4 5 3 1 -1	5 17
5 6 1 2 1 1 3 2 2 4 3 3 4 4 3 5 5 4 5 6 1 3 5 2 1 -1	2 7
5 6 1 2 6 1 3 5 1 4 4 2 3 3 2 4 2 3 4 1 1 3 4 2 5 1 -1	3 8 
6 10 1 2 1 1 3 1 2 4 1 2 6 1 3 2 1 3 5 1 3 6 1 5 2 1 5 4 1 6 5 1 1 5 2 6 5 3 1 2 3 4 -1	0 0