

แข่งรถ

ทางเมืองพัทยาต้องการจัดการแข่งรถโอลิมปิกนานาชาติ (The International Olympiad in Racing: IOR) 2011 ควบคู่ไปกับการแข่งขัน IOI ซึ่งในฐานะเจ้าภาพเราจำเป็นต้องหาเส้นทางที่ดีที่สุดเท่าที่จะเป็นไปได้เพื่อการแข่งรถนี้

ในพื้นที่ของเมืองพัทยา-จังหวัดชลบุรี มีอำเภอทั้งหมด N อำเภอ เชื่อมต่อกันด้วยเครือข่ายถนนไฮเวย์จำนวน $N-1$ เส้น ถนนไฮเวย์แต่ละเส้นเป็นถนนแบบสองทิศทางซึ่งเชื่อมต่ออำเภอสองอำเภอที่แตกต่างกัน และไฮเวย์เหล่านี้มีความยาวเป็นตัวเลขจำนวนเต็ม มีหน่วยเป็นกิโลเมตร นอกจากนี้ระหว่างอำเภอคูใด ๆ ยังมีเส้นทางที่เป็นไปได้เพียงเส้นทางเดียวเท่านั้น นั่นคือ จะมีเส้นทางเดียวเท่านั้นในการเดินทางจากอำเภอหนึ่งไปยังอีกอำเภอหนึ่ง โดยผ่านชุดลำดับของไฮเวย์ชุดหนึ่งซึ่งผ่านอำเภอแต่ละอำเภอเพียงครั้งเดียวเท่านั้น

ในการแข่ง IOR มีกฎข้อพิเศษที่บังคับว่า เส้นทางที่ใช้จะต้องมีความยาวรวมเท่ากับ K กิโลเมตรเท่านั้น โดยเริ่มต้นและสิ้นสุดที่เมืองคนละเมืองกัน จากข้อกำหนดนี้จะเห็นได้อย่างชัดเจนว่า บนเส้นทางเดียวกันจะไม่มีการใช้ถนนไฮเวย์เส้นใดเส้นหนึ่ง (รวมถึงจะไม่ผ่านอำเภอใด ๆ) สองครั้งขึ้นไป เพื่อป้องกันรถชนกัน และเพื่อให้เกิดการรบกวนการจราจรน้อยที่สุด เส้นทางที่เลือกนี้จะต้องเลือกโดยใช้ถนนไฮเวย์จำนวนน้อยเส้นที่สุดเท่าที่จะเป็นไปได้

งานของคุณ

เขียนโปรแกรมย่อย `best_path(N,K,H,L)` ซึ่งรับพารามิเตอร์ดังต่อไปนี้

- N – เป็นจำนวนอำเภอ ซึ่งมีตัวเลขกำกับตั้งแต่ 0 ถึง $N-1$
- K – ความยาวรวมของเส้นทางแข่ง
- H – อาร์เรย์สองมิติเก็บข้อมูลไฮเวย์ โดยสำหรับค่า i ที่มีค่า $0 \leq i < N-1$ ไฮเวย์ i เชื่อมต่อระหว่างอำเภอ $H[i][0]$ และ $H[i][1]$
- L – เป็นอาร์เรย์หนึ่งมิติเก็บความยาวของถนนไฮเวย์ โดยสำหรับค่า i ที่มีค่า $0 \leq i < N-1$ นั้น $L[i]$ เป็นความยาวของไฮเวย์ i

คุณอาจจะสมมติว่า ค่าทั้งหมดในอาร์เรย์ H มีค่าอยู่ตั้งแต่ 0 ถึง $N-1$ รวมหัวท้าย และสมมติว่า ไฮเวย์ที่เก็บในอาร์เรย์นี้เชื่อมต่ออำเภอทุกอำเภอตามข้างต้น และอาจจะสมมติว่า ค่าในอาร์เรย์ L เป็นตัวเลขจำนวนเต็มตั้งแต่ 0 ถึง 1,000,000 รวมหัวท้าย

โปรแกรมย่อยของคุณจะต้องคืนค่า จำนวนไฮเวย์ที่น้อยที่สุด ที่ประกอบกันเป็นเส้นทางแข่งที่ถูกต้องซึ่งมีความยาว K กิโลเมตรพอดี และถ้าไม่มีเส้นทางดังกล่าว โปรแกรมย่อยของคุณจะต้องคืนค่า -1

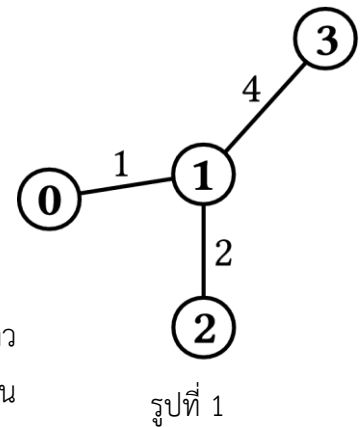
ตัวอย่าง

ตัวอย่างที่ 1

จากตัวอย่างที่แสดงในรูปที่ 1 กำหนดให้ $N=4$, $K=3$,

	0	1		1
H=	1	2	L=	2
	1	3		4

เส้นทางที่เป็นไปได้ เริ่มจากอำเภอ 0 ไปยังอำเภอ 1 และสิ้นสุดที่อำเภอ 2 ความยาวของเส้นทางนี้จะเท่ากับ 1 กิโลเมตร + 2 กิโลเมตร = 3 กิโลเมตร พอดี และในเส้นทางนี้ประกอบด้วยถนนไฮเวย์สองเส้น เส้นทางนี้เป็นเส้นทางที่ดีที่สุดเท่าที่จะเป็นไปได้ ดังนั้น $\text{best_path}(N,K,H,L)$ จะต้องคืนค่า 2 กลับไป

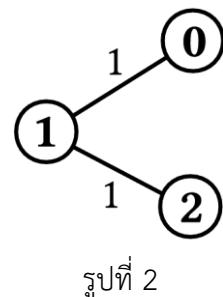


ตัวอย่างที่ 2

จากตัวอย่างที่แสดงในรูปที่ 2 กำหนดให้ $N=3$, $K=3$,

	0	1		1
H=	1	2	L=	1

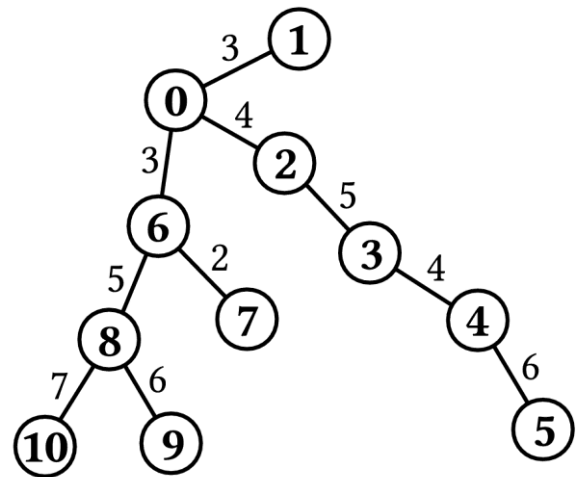
ในตัวอย่างนี้ ไม่มีคำตอบที่ถูกต้อง ในกรณีนี้ $\text{best_path}(N,K,H,L)$ ต้องคืนค่า -1



ตัวอย่างที่ 3

จากตัวอย่างที่แสดงในรูปที่ 3 กำหนดให้ $N=11$, $K=12$,

0 1	3
0 2	4
2 3	5
3 4	4
4 5	6
0 6	3
6 7	2
6 8	5
8 9	6
8 10	7



รูปที่ 3

เส้นทางเส้นหนึ่งที่เป็นไปได้ ประกอบด้วยถนนไฮเวย์ 3 เส้น จากอำเภอ 6 ไปยังอำเภอ 0 ไปยังอำเภอ 2 และไปยังอำเภอ 3 และยังมีเส้นทางอีกเส้นหนึ่งที่เริ่มต้นที่อำเภอ 10 ผ่านอำเภอ 8 ไปยังอำเภอ 6 ทั้งสองเส้นทางนี้มีระยะทางเท่ากับ 12 กิโลเมตร พอติดตามโจทย์กำหนด แต่เส้นทางที่สองเป็นเส้นทางที่ดีที่สุด เพราะไม่มีเส้นทางเส้นทางที่ถูกต้องอื่นใดซึ่งประกอบด้วยไฮเวย์เพียงเส้นทางเดียว ดังนั้น $\text{best_path}(N,K,H,L)$ ต้องคืนค่า 2

งานย่อย

งานย่อย 1 (9 คะแนน)

- $1 \leq N \leq 100$
- $1 \leq K \leq 100$
- เครือข่ายของไฮเวย์เรียงตัวกันเป็นเส้นตรง กล่าวคือ สำหรับ i ทุกค่าที่ $0 \leq i < N-1$ ไฮเวย์ i เชื่อมต่อเมือง i และเมือง $i+1$

งานย่อย 2 (12 คะแนน)

- $1 \leq N \leq 1\,000$
- $1 \leq K \leq 1\,000\,000$

งานย่อย 3 (22 คะแนน)

- $1 \leq N \leq 200\,000$
- $1 \leq K \leq 100$

งานย่อย 4 (57 คะแนน)

- $1 \leq N \leq 200\,000$
- $1 \leq K \leq 1\,000\,000$

รายละเอียดการเขียนโปรแกรม

ข้อจำกัด

- เวลาที่ CPU ทำงาน: 3 วินาที
- หน่วยความจำ: 256 MB

หมายเหตุ: เราไม่ได้กำหนดข้อจำกัดของขนาดของหน่วยความจำ stack เนื่องจากขนาดของหน่วยความจำ stack รวมอยู่ในขนาดหน่วยความจำทั้งหมดที่โปรแกรมใช้อยู่แล้ว

อินเตอร์เฟส (API)

- โพลเดอร์ของปัญหา: race/
- ไฟล์ที่ผู้แข่งขันต้องแก้ไข: race.c หรือ race.cpp หรือ race.pas
- อินเตอร์เฟสที่ผู้แข่งขันใช้: race.h หรือ race.pas
- อินเตอร์เฟสที่ตัวตรวจใช้: race.h หรือ racelib.pas
- ตัวตรวจตัวอย่าง: grader.c หรือ grader.cpp หรือ grader.pas
- ไฟล์ข้อมูลนำเข้าของตัวตรวจตัวอย่าง: grader.in.1, grader.in.2, ...

หมายเหตุ: ตัวตรวจตัวอย่างอ่านไฟล์ที่มีรูปแบบดังต่อไปนี้

- บรรทัดที่ 1: N และ K
 - บรรทัดที่ 2 ถึง N : ข้อมูลไฮเวย์ กล่าวคือ บรรทัดที่ $i+2$ มี $H[i][0]$, $H[i][1]$, $L[i]$ คั่นด้วยช่องว่างหนึ่งช่อง สำหรับ i ทุกค่าที่ $0 \leq i < N-1$
 - บรรทัดที่ $N+1$: คำตอบที่ถูกต้อง
 - ไฟล์ข้อมูลส่งออกของตัวตรวจ: grader.expect.1, grader.expect.2, ...
- สำหรับงานนี้ ไฟล์ข้างต้นแต่ละไฟล์จะมีข้อความ “Correct.” เท่านั้น