การแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ ครั้งที่ 20

ณ มหาวิทยาลัยศิลปากร วิทยาเขตพระราชวังสนามจันทร์

ข้อสอบข้อที่ 2 จากทั้งหมด 3 ข้อ

วันพุธที่ 15 พฤษภาคม 2567 เวลา 8.00 - 13.00 น.

เส้นทาง (Route)

เนื่องด้วยมหาวิทยาลัยศิลปากรมีชื่อเสียงโด่งดังจนทำให้มีนักศึกษาเข้ามาเรียนต่อในระดับอุดมศึกษาเป็นจำนวนมาก จนเกิดปัญหาเรื่องสวัสดิการรถรับส่งภายในมหาวิทยาลัยมีปริมาณไม่เพียงพอ สมหวังได้รับการว่าจ้างให้มาออกแบบระบบขนส่งระหว่างสถานที่สำคัญต่าง ๆ ในมหาวิทยาลัยศิลปากร ซึ่งมีจุดสำคัญทั้งหมด N จุด ทั้งนี้เพื่อความสะดวกแต่ละจุดดังกล่าวจะถูกกำกับด้วยหมายเลข 1 ถึง N สมหวังต้องดูแลรถขนส่งรุ่นใหม่ที่เป็นระบบไฟฟ้าทั้งหมดจำนวน M คัน และเพื่อความสะดวกเช่นเคย รถแต่ละคันดังกล่าวจะถูกกำกับด้วยหมายเลข 1 ถึง M ด้วยข้อกำหนดด้านความปลอดภัย รถขนส่งแต่ละคันจะหยุดเพื่อรับส่งผู้โดยสารระหว่างจุดสำคัญที่กำหนด 2 จุดเท่านั้น ไม่อนุญาตให้แวะจอดระหว่างทางโดยเด็ดขาด ทั้งนี้รถแต่ละคันจะมีค่าใช้จ่ายประจำรถ รถหมายเลข i มีค่าใช้จ่ายเฉพาะคันคือ w_i เมื่อ $i = 1, \dots, M$

เมื่อสมหวังได้ปรึกษาผู้เชี่ยวชาญทางด้านทฤษฎีกราฟ ก็พบว่าจริง ๆ แล้ว หากเราทราบว่ารถคันไหนวิ่งระหว่างจุดไหน เราสามารถคำนวณ **"การจัดสรรแบบประหยัด"** ได้โดย "การจัดสรรแบบประหยัด" คือ การเลือกรถ $N - 1$ คันจากเส้นทางที่กำหนดให้ เพื่อให้มีค่าใช้จ่ายรวมของรถที่เลือกน้อยที่สุด แต่ผู้ใช้บริการยังสามารถเดินทางไปมาระหว่างจุดทั้ง N จุดได้อยู่

อย่างไรก็ตาม เรายังไม่ทราบว่ารถคันไหนวิ่งระหว่างจุดไหน สมหวังต้องการความช่วยเหลือจากคุณที่จะช่วยออกแบบการจัดสรรเส้นทางของระบบขนส่งที่มีการใช้รถทั้ง M คันในการให้บริการรับส่งระหว่างจุดสำคัญทั้ง N จุด โดยออกแบบให้ "การจัดสรรแบบประหยัด" ของเส้นทางที่คุณกำหนดนั้น มีผลรวมของค่าใช้จ่ายของรถ $N - 1$ คันที่อยู่ใน "การจัดสรรแบบประหยัด" นั้นมีค่า**มากที่สุด**

งานของคุณ

จงเขียน**ฟังก์ชันภาษา C++** เพื่อสร้างเวกเตอร์ที่มีขนาด M ระบุเส้นทางเดินรถของรถคันที่ i ($i = 1, \dots, M$)

ข้อนี้มีรูปแบบการเขียนโปรแกรมที่แตกต่างจากรูปแบบปรกติ ซึ่งจะมีใช้ในการแข่งขันจริง โปรดดูส่วนท้ายของโจทย์ข้อนี้เกี่ยวกับวิธีการใช้งาน

รายละเอียดการเขียนโปรแกรม

ผู้เข้าแข่งขันจะต้องเขียนฟังก์ชันต่อไปนี้

```
std::vector<pair<int,int>> route(int N, std::vector<int> W)
```

โดยที่พารามิเตอร์ต่าง ๆ คือ

- N คือจำนวนสถานที่สำคัญในมหาวิทยาลัย
- W คือค่าใช้จ่ายของรถขนส่งแต่ละคัน
    - `W.size()` จะมีค่าเท่ากับ M
    - รับประกันว่าข้อมูลในเวกเตอร์ `W` **เรียงจากน้อยไปมาก**

เมื่อจบการทำงานของฟังก์ชันนี้แล้ว โปรแกรมของผู้เข้าแข่งขันต้องกำหนดเส้นทางวิ่งให้กับรถขนส่งแต่ละคันผ่านค่าที่คืนมาโดยฟังก์ชันนี้ โดยฟังก์ชันนี้จะต้องคืนค่าเวกเตอร์ขนาด M ของ `pair` ซึ่งระบุว่ารถขนส่งแต่ละคันวิ่งเชื่อมระหว่างสถานที่สำคัญหมายเลขใดไปยังหมายเลขใด เพื่อความสะดวก ถ้ากำหนดให้ A คือเวกเตอร์ที่คืนค่ามา A จะต้องเป็นไปตามเงื่อนไขต่อไปนี้

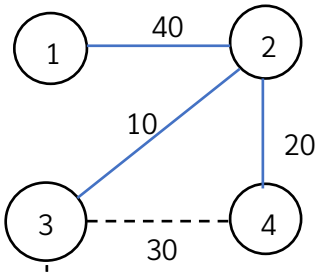
- `A[`i`]` จะเป็นคู่หมายเลขสถานที่สำคัญสำหรับรถขนส่งที่มีค่าใช้จ่ายเป็น `W[`i`]` กล่าวคือ จะมีเส้นทางรถขนส่งที่วิ่งระหว่างสถานที่สำคัญหมายเลข `A[`i`].first` กับ `A[`i`].second` โดยรถขนส่งคันดังกล่าวมีค่าใช้จ่ายเป็น `W[`i`]`
- เส้นทางที่พิจารณาจาก A ต้องทำให้เป็นไปได้ที่จะเลือกเส้นทางรถขนส่งเพียง $N-1$ เส้นทางที่ทำให้สามารถเดินทางระหว่างจุดสำคัญทุกจุดได้
- ต้องไม่มี i,j ใด ๆ ที่ทำให้ `A[`i`]==A[`j`]` (กล่าวคือ ต้องไม่มีเส้นทางใดมีรถขนส่งวิ่งมากกว่า 1 คัน)
- ต้องไม่มี i ใด ๆ ที่ `A[`i`].first = A[`i`].second` (กล่าวคือต้องไม่มีเส้นทางใดที่รถขนส่งวิ่งวนณ จุดเดิมโดยไม่ไปจุดอื่น)

หากมีเวกเตอร์หลายรูปแบบที่ตรงกับเงื่อนไขที่โจทย์กำหนด ผู้เข้าแข่งขันสามารถเลือกตอบแบบใดแบบหนึ่งที่ตรงตามเงื่อนไขก็ได้

ตัวอย่างที่ 1

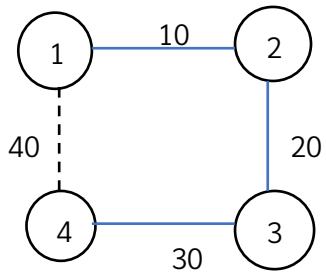
กำหนดให้ $N = 4$ และ W เป็น [10, 20, 30, 40] ซึ่งหมายความว่ามีจุดสำคัญ 4 จุด และมีรถขนส่ง 4 คัน ($M = 4$) ซึ่งมีค่าใช้จ่ายคันที่ 1, 2, 3 และ 4 เป็น 10, 20, 30 และ 40 ตามลำดับ ตัวอย่างหนึ่งของคำตอบที่ดีที่สุดของข้อนี้คือการคืนค่า [{2,3}, {2,4}, {3,4}, {2,1}] ซึ่งหมายถึงการกำหนดรถขนส่งประจำเส้นทางดังนี้

- รถขนส่งคันที่ 1 วิ่งรับส่งระหว่างจุด 2 กับ 3 (ค่าใช้จ่าย 10)
- รถขนส่งคันที่ 2 วิ่งรับส่งระหว่างจุด 2 กับ 4 (ค่าใช้จ่าย 20)
- รถขนส่งคันที่ 3 วิ่งรับส่งระหว่างจุด 3 กับ 4 (ค่าใช้จ่าย 30)
- รถขนส่งคันที่ 4 วิ่งรับส่งระหว่างจุด 2 กับ 1 (ค่าใช้จ่าย 40)



ภาพที่ 1: ภาพเส้นทางประกอบตัวอย่างที่ 1

ซึ่งสามารถวาดเป็นแผนภาพได้ดังภาพที่ 1 สำหรับรูปแบบเส้นทางการเดินรถดังกล่าวสามารถลดจำนวนรถขนส่งให้เหลือเพียง 3 คันได้ โดยหากลดการใช้งานรถขนส่งคันที่ 3 จะทำให้ยังสามารถเดินทางไปยังจุดสำคัญทั้ง 4 จุดได้ โดยมีค่าใช้จ่าย 70 หน่วย และเป็นการออกแบบที่เป็น **“การจัดสรรแบบประหยัด”**



ทั้งนี้ หากฟังก์ชันของเราคืนค่ากลับมาเป็น [{1,2}, {2,3}, {3,4}, {4,1}] ซึ่งหมายถึงการกำหนดเส้นทางดังภาพที่ 2 เมื่อลดจำนวนรถขนส่งเหลือเพียง 3 คัน คือใช้รถคันที่ 1, 2 และ 3 จะเป็นการออกแบบที่เป็น **"การจัดสรรแบบประหยัด"** โดยมีค่าใช้จ่าย 10+20+30=60 หน่วย

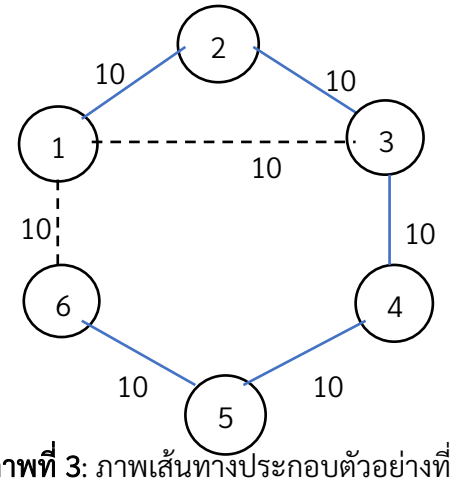
สำหรับตัวอย่างที่ 1 นี้ เมื่อพิจารณาเส้นทางจากภาพที่ 1 และภาพที่ 2 จะได้ว่า ภาพที่ 1 **เป็น** "การจัดสรรแบบประหยัด" ที่มีค่า**มากที่สุด**
แต่ ภาพที่ 2 **เป็น** "การจัดสรรแบบประหยัด" **แต่ไม่ได้มีค่ามากที่สุด**

ภาพที่ 2: ภาพเส้นทางประกอบตัวอย่างที่ 1

(หากต้องการทดสอบโปรแกรมด้วยตัวอย่างนี้ ให้ดูในหัวข้อ "คำอธิบายไฟล์เกรดเดอร์ด้านท้ายของโจทย์")

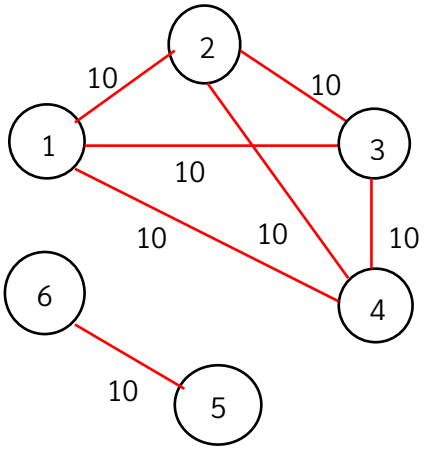
ตัวอย่างที่ 2

กำหนดให้ค่า N = 6 และ W เป็น [10, 10, 10, 10, 10, 10, 10] ซึ่งหมายความว่ามีจุดสำคัญ 6 จุด และมีรถขนส่ง 7 คัน (M = 7) ซึ่งรถขนส่งแต่ละคันมีค่าใช้จ่ายเป็น 10 หน่วยทุกคัน ตัวอย่างหนึ่งของคำตอบที่ดีที่สุดของข้อนี้คือการคืนค่า [{1,2}, {2,3}, {3,4}, {4,5}, {5,6}, {6,1}, {1,3}] ซึ่งหมายถึงหากใช้รถขนส่งเพียง 5 คัน ถือว่าเป็น "การจัดสรรแบบประหยัด" ที่มีค่า**มากที่สุด** โดยมีค่าใช้จ่าย 50 หน่วย



ภาพที่ 3: ภาพเส้นทางประกอบตัวอย่างที่ 2

อย่างไรก็ตาม หากฟังก์ชันของเราคืนค่ามาเป็น [{1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}, {5,6}] ซึ่งหมายถึงการกำหนดเส้นทางดังภาพที่ 4 นี้ ให้สังเกตว่า การกำหนดเส้นทางดังกล่าวไม่ตรงกับเงื่อนไขที่กำหนดให้ เนื่องจากเราไม่สามารถเลือกเส้นทางจำนวน N-1 เส้นทางจากเส้นทางที่กำหนดให้ที่ทำให้เราสามารถเดินทางไปมาระหว่างสถานที่สำคัญทั้งหมดได้



ภาพที่ 4: ภาพเส้นทางประกอบตัวอย่างที่ 2

(หากต้องการทดสอบโปรแกรมด้วยตัวอย่างนี้ ให้ดูในหัวข้อ "คำอธิบายไฟล์เกรดเดอร์ด้านท้ายของโจทย์")

ขอบเขตของข้อมูล

- $2 \leq N \leq 300{,}000$
- $N-1 \leq M \leq \min\left(1{,}000{,}000\ , \frac{N(N-1)}{2}\right)$

ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลนำเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล	1 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล	512 MB
คะแนนสูงสุดของโจทย์	100 คะแนน

ข้อมูลเพิ่มเติมเกี่ยวกับชุดทดสอบ

ข้อมูลแนะนำที่เกี่ยวข้องกับชุดทดสอบ มีดังนี้

กลุ่มชุดทดสอบที่	คะแนนสูงสุดของกลุ่มชุดทดสอบนี้	เงื่อนไข
1	10	$\mathrm{M} = \mathrm{N}$
2	15	$\mathrm{M} \leq N + 1$
3	20	$\mathrm{N} \leq 6$
4	20	$\mathrm{N} \leq 200$
5	35	ไม่มีเงื่อนไขอื่น

การคิดคะแนน

ในข้อนี้ ผู้เข้าแข่งขันสามารถได้คะแนนโดยที่**ไม่จำเป็นต้องตอบคำตอบที่ดีที่สุด**ก็ได้ โดยคะแนนที่ได้จะคิดจากวิธีการดังนี้

1. ในชุดทดสอบแต่ละชุด การกำหนดเส้นทางของผู้เข้าแข่งขันจะต้องตรงตามเงื่อนไขในส่วน “รายละเอียดการเขียนโปรแกรม” ระบบถึงจะทำการคิดคะแนนให้ หากไม่ตรงตามเงื่อนไข (เช่น กำหนดเส้นทางที่ไม่สามารถเดินทางระหว่างสถานที่สำคัญได้ทั้งหมด หรือมีการกำหนดเส้นทางซ้ำกันให้กับรถขนส่ง) คะแนนจะเป็น 0 ในชุดทดสอบดังกล่าว

2. ในกรณีที่การกำหนดเส้นทางตรงตามเงื่อนไข กำหนดให้

   - S คือผลต่างระหว่าง “ค่าใช้จ่ายรวม” กับ ค่าใช้จ่ายของ “การจัดสรรแบบประหยัด” ที่มีค่ามากที่สุด ที่สามารถทำได้ด้วยโปรแกรมของ**<u>กรรมการออกข้อสอบของการแข่งขัน</u>** โดยที่ “ค่าใช้จ่ายรวม” หมายถึง $\sum_{i=1}^{M} w_i$
   - T คือผลต่างระหว่าง “ค่าใช้จ่ายรวม” กับ ค่าใช้จ่ายของ “การจัดสรรแบบประหยัด” ที่มีค่ามากที่สุด ที่สามารถทำได้ด้วยโปรแกรมของ**<u>ผู้เข้าแข่งขัน</u>**

     คะแนนที่ได้จะเป็นดังนี้
   - ถ้า $S \geq T$ ผู้เข้าแข่งขันจะได้คะแนนเต็ม 100 ในชุดทดสอบนั้น
   - ถ้า $S < T$ ผู้เข้าแข่งขันจะได้คะแนนเป็น $\frac{S^2}{T^2} \times 100$

คำอธิบายไฟล์เกรดเดอร์

ไฟล์เกรดเดอร์จะรับข้อมูลนำเข้าสองบรรทัดในรูปแบบต่อไปนี้

- บรรทัดที่ 1: รับค่า N และ M
- บรรทัดที่ 2: รับจำนวนเต็ม M ตัวซึ่งคือค่า W_1 ถึง W_M

หลังจากนั้นเกรดเดอร์จะเรียกฟังก์ชัน `route` ของผู้เข้าแข่งขันแล้วแสดงผลลัพธ์ที่ได้จากฟังก์ชันออกทางหน้าจอ โดยแสดงข้อมูลออกมา M บรรทัด แต่ละบรรทัดจะระบุถึงเส้นทางการเดินรถแต่ละคัน และ ในแต่ละบรรทัดประกอบด้วยตัวเลข 3 ตัวคือ u v และ w ซึ่งคือหมายเลขของสถานที่สำคัญที่รถบัสคันนั้นวิ่งไปมาและค่าใช้จ่ายของรถบัสคันดังกล่าว

ตัวอย่างข้อมูลนำเข้าและข้อมูลส่งออกเมื่อใช้ไฟล์เกรดเดอร์

ตัวอย่างที่	ข้อมูลนำเข้า	ข้อมูลส่งออก
1 (ตัวอย่างนี้ตรงกับ “ตัวอย่างที่ 1” ข้างต้น)	4 4 10 20 30 40	2 3 10 2 4 20 3 4 30 1 2 40
2 (ตัวอย่างนี้ตรงกับ “ตัวอย่างที่ 2” ข้างต้น)	6 7 10 10 10 10 10 10 10	1 2 10 2 3 10 3 4 10 4 5 10 5 6 10 6 1 10 1 3 10

คำแนะนำในการเขียนโปรแกรม

สำหรับข้อนี้ วิธีการเขียนโปรแกรมและส่งโปรแกรมเข้าสู่ระบบ grader จะเป็นรูปแบบที่แตกต่างจากรูปแบบทั่วไปของการเขียนโปรแกรม ที่ผู้เข้าแข่งขันจะต้องเขียนโปรแกรมในส่วนรับข้อมูลนำเข้า และส่วนแสดงผล โดยในข้อนี้ ผู้เข้าแข่งขันจะต้องเขียนเฉพาะส่วนการคำนวณที่จำเป็นลงในฟังก์ชันที่กำหนดให้ โดยข้อมูลนำเข้าต่าง ๆ จะสามารถใช้ได้ผ่านพารามิเตอร์ของฟังก์ชัน และผลการคำนวณจะต้องส่งกลับผ่านฟังก์ชันด้วยเช่นกัน โดยที่ผู้เข้าแข่งขันไม่ต้องเขียนโปรแกรมในส่วนรับข้อมูลนำเข้า และไม่ต้องเขียนส่วนแสดงผล

ผู้เข้าแข่งขันจะได้รับไฟล์ต่าง ๆ หลายไฟล์ดังนี้ ผู้เข้าแข่งขันจะต้องแก้ไขไฟล์เดียวตามที่โจทย์กำหนด และส่งไฟล์ดังกล่าวเข้าสู่ระบบ

- grader.cpp เป็นไฟล์หลักที่ทำหน้าที่รับข้อมูลนำเข้าและส่งออก
- route.cpp เป็นไฟล์ที่ผู้เข้าแข่งขันจะต้องเขียนโปรแกรมคำนวณต่าง ๆ และเป็นไฟล์ที่ผู้เข้าแข่งขันส่งเข้าสู่ระบบ
- run_cpp.sh เป็นไฟล์ที่ผู้เข้าแข่งขันสามารถเรียกใช้เพื่อทำการรันโปรแกรมทั้งหมดได้
- compile_cpp.sh เป็นไฟล์ที่ผู้เข้าแข่งขันสามารถเรียกใช้เพื่อทำการคอมไพล์โปรแกรมได้
- route.h เป็นไฟล์ประกอบการคอมไพล์ ห้ามผู้เข้าแข่งขันแก้ไขไฟล์นี้

ผู้เข้าแข่งขันจะได้รับไฟล์ .zip ซึ่งสามารถดาวโหลดได้จากระบบโดยในไฟล์ดังกล่าวเมื่อขยายออกมาแล้วจะมีไฟล์ทั้ง 4 อยู่ ผู้เข้าแข่งขันควรใช้ text editor หรือ IDE ในการแก้ไขไฟล์ route.cpp และทำการคอมไพล์หรือรันโปรแกรมผ่านการเรียกใช้ฟังก์ชัน compile_cpp.sh หรือ run_cpp.sh

เมื่อผู้เข้าแข่งขันสามารถเรียกไฟล์ run_cpp.sh เพื่อทำการรันโปรแกรม โปรแกรมจะทำงานตามที่ได้ระบุไว้ในไฟล์ grader.cpp ซึ่งหัวข้อ “คำอธิบายไฟล์เกรดเดอร์” อธิบายถึงการทำงานของไฟล์ดังกล่าว

ข้อบังคับการเขียนโปรแกรม

- ผู้เข้าแข่งขันต้องส่งเฉพาะไฟล์ route.cpp เท่านั้น
- ในไฟล์ route.cpp ต้องไม่มีฟังก์ชัน main
- ในไฟล์ route.cpp ต้องไม่มีการรับข้อมูลหรือส่งข้อมูลผ่าน cin cout printf scanf