

Road Closures

ในเมืองสุราบายา มีแยกอยู่ N แยก มีหมายเลขตั้งแต่ 0 ถึง $N - 1$ แยกเหล่านี้เชื่อมกันด้วยถนนที่ไปได้สองทางจำนวน $N - 1$ เส้น มีหมายเลข 0 ถึง $N - 1$ โดยที่รับประกันว่ามีเส้นทางเส้นทางหนึ่งเส้นพอดีในการเดินทางระหว่างคู่ของแยกใด ๆ ผ่านถนนเหล่านี้ ถนน i ($0 \leq i \leq N - 2$) เชื่อมแยก $U[i]$ กับแยก $V[i]$

เพื่อสร้างความตระหนักด้านสิ่งแวดล้อม ปก เด็กเล็ก ผู้ว่าการจังหวัดสุราบายาจึงวางแผนจะจัดงาน Car Free Day เพื่อจะกระตุ้นกิจกรรมนี้ ปก เด็กเล็กจะจัดการปิดถนน ปก เด็กเล็กจะเริ่มโดยการเลือกจำนวนเต็มที่ไม่เป็นลบ k จากนั้นจะปิดถนนโดยรับประกันว่าทุก ๆ แยกจะต้องติดโดยตรงกับถนนที่ไม่ปิดจำนวนไม่เกิน k เส้น ค่าใช้จ่ายในการปิดถนน i คือ $W[i]$

ช่วยปก เด็กเล็กหาว่าจะต้องใช้ค่าใช้จ่ายน้อยที่สุดในการปิดถนนสำหรับทุก ๆ ค่า k ที่เป็นไปได้ ($0 \leq k \leq N - 1$)

Implementation Details

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
int64[] minimum_closure_costs(int N, int[] U, int[] V, int[] W)
```

- N : จำนวนแยกในสุราบายา
- U และ V : อาร์เรย์ขนาด $N - 1$ โดยที่แยก $U[i]$ และ $V[i]$ เชื่อมด้วยถนน i .
- W : อาร์เรย์ขนาด $N - 1$ โดยที่ $W[i]$ คือค่าใช้จ่ายในการปิดถนน i .
- ฟังก์ชันนี้จะต้องคืนอาร์เรย์ขนาด N โดยที่ สำหรับทุก ๆ ค่า k ($0 \leq k \leq N - 1$), ข้อมูลตัวที่ k ในอาร์เรย์คือค่าใช้จ่ายน้อยที่สุดในการปิดถนนเพื่อรับประกันว่าทุก ๆ แยกจะติดกับถนนที่ไม่ปิดจำนวนไม่เกิน k เส้น
- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้งพอดี

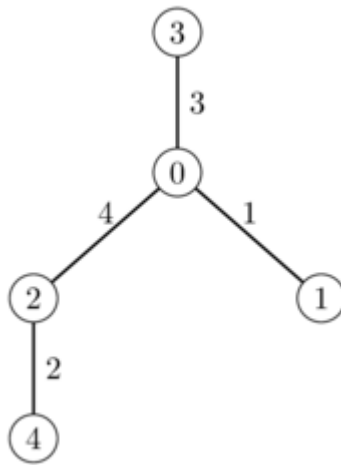
Examples

Example 1

พิจารณาตัวอย่างการเรียกต่อไปนี้:

```
minimum_closure_costs(5, [0, 0, 0, 2], [1, 2, 3, 4], [1, 4, 3, 2])
```

นี้หมายความว่าในเมืองจะมีแยกจำนวน 5 แยก และมีถนน 4 เส้นที่เชื่อมคู่ของแยกต่อไปนี้ (0,1), (0,2), (0,3), และ (2,4) ด้วยค่าใช้จ่ายในการปิดถนน 1, 4, 3, และ 2, ตามลำดับ



วิธีที่จะเสียค่าใช้จ่ายน้อยที่สุดเป็นดังนี้

- ถ้าปิด เคเบิลเลือก $k = 0$, ทำให้ต้องปิดทุกถนน ด้วยค่าใช้จ่ายเท่ากับ $1 + 4 + 3 + 2 = 10$
- ถ้าปิด เคเบิลเลือก $k = 1$, ถนนที่จะต้องปิดคือถนน 0 และถนน 1 โดยมีค่าใช้จ่ายเท่ากับ $1 + 4 = 5$
- ถ้าปิด เคเบิลเลือก $k = 2$, จะต้องปิดถนน 0 ด้วยค่าใช้จ่าย 1
- ถ้าปิด เคเบิลเลือก $k = 3$ หรือ $k = 4$, ไม่จำเป็นต้องปิดถนนใด ๆ เลย

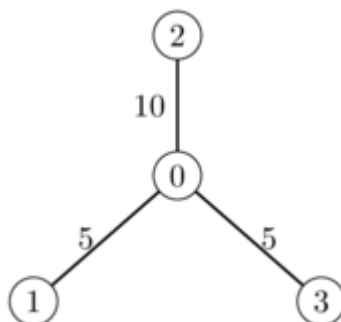
ดังนั้น ฟังก์ชัน `minimum_closure_costs` จะต้องคืนค่า `[10, 5, 1, 0, 0]`

Example 2

พิจารณาตัวอย่างการเรียกต่อไปนี้:

```
minimum_closure_costs(4, [0, 2, 0], [1, 0, 3], [5, 10, 5])
```

นี้หมายความว่าในเมืองจะมีแยกจำนวน 4 แยก และมีถนน 3 เส้นที่เชื่อมคู่ของแยกต่อไปนี้ $(0, 1)$, $(2, 0)$, และ $(0, 3)$ ด้วยค่าใช้จ่ายในการปิดถนน 5, 10, และ 5, ตามลำดับ



วิธีที่จะเสียค่าใช้จ่ายน้อยที่สุดเป็นดังนี้

- ถ้าปิด เคเบิลเลือก $k = 0$, ทำให้ต้องปิดทุกถนน ด้วยค่าใช้จ่ายเท่ากับ $5 + 10 + 5 = 20$
- ถ้าปิด เคเบิลเลือก $k = 1$, จะต้องปิดถนน 0 และถนน 2 ด้วยค่าใช้จ่ายเท่ากับ $5 + 5 = 10$
- ถ้าปิด เคเบิลเลือก $k = 2$, จะสามารถเลือกปิดถนน 0 หรือถนน 2 ก็ได้ ด้วยค่าใช้จ่ายเท่ากับ 5

- ถ้าปิด เคสเลือก $k = 3$, ไม่จำเป็นต้องปิดถนนใด ๆ เลย

ดังนั้น ฟังก์ชัน `minimum_closure_costs` จะต้องคืนค่า `[20, 10, 5, 0]`

Constraints

- $2 \leq N \leq 100\,000$
- $0 \leq U[i], V[i] \leq N - 1$ (สำหรับทุก $0 \leq i \leq N - 2$)
- สามารถเดินทางระหว่างคู่ของแยกใด ๆ ได้ โดยใช้ถนนเหล่านี้
- $1 \leq W[i] \leq 10^9$ (สำหรับทุก $0 \leq i \leq N - 2$)

Subtasks

1. (5 points) $U[i] = 0$ (สำหรับทุก $0 \leq i \leq N - 2$)
2. (7 points) $U[i] = i, V[i] = i + 1$ (สำหรับทุก $0 \leq i \leq N - 2$)
3. (14 points) $N \leq 200$
4. (10 points) $N \leq 2000$
5. (17 points) $W[i] = 1$ (สำหรับทุก $0 \leq i \leq N - 2$)
6. (25 points) $W[i] \leq 10$ (สำหรับทุก $0 \leq i \leq N - 2$)
7. (22 points) ไม่มีเงื่อนไขใด ๆ เพิ่มเติม

Sample Grader

เกรดเดอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้:

- line 1: N
- line $2 + i$ ($0 \leq i \leq N - 2$): $U[i] \ V[i] \ W[i]$

เกรดเดอร์ตัวอย่างพิมพ์ผลลัพธ์หนึ่งบรรทัดเป็นอาร์เรย์ที่คืนจากฟังก์ชัน `minimum_closure_costs`