

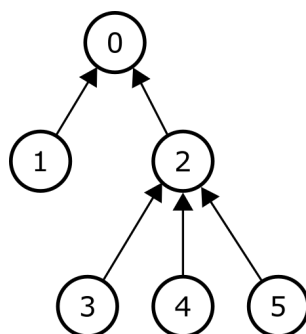
สัมประสิทธิ์พหุคูณ

พิจารณา ต้นไม้ ซึ่งประกอบด้วย จุดยอด N จุด มีตัวเลขตั้งแต่ 0 ถึง $N - 1$ จุดยอด 0 เราเรียกว่าเป็น **ราก** จุดยอดทุกจุด ยกเว้นราก ล้วนมี **พ่อ** หนึ่งจุด สำหรับทุกๆ i โดยที่ $1 \leq i < N$ พ่อของจุดยอด i คือจุดยอด $P[i]$ โดยที่ $P[i] < i$ ทั้งนี้เราถือว่า $P[0] = -1$

สำหรับทุกๆ จุดยอด i ($0 \leq i < N$) **ต้นไม้ย่อย** ของจุดยอด i คือเซตของจุดยอดต่อไปนี้:

- จุดยอด i และ
- จุดยอดใดก็ตามที่มีพ่อเป็นจุดยอด i และ
- จุดยอดใดก็ตามที่มีพ่อของพ่อเป็นจุดยอด i และ
- จุดยอดใดก็ตามที่มีพ่อของพ่อของพ่อเป็นจุดยอด i และ
- ฯลฯ

ภาพด้านล่างนี้แสดงต้นไม้ตัวอย่างที่มี $N = 6$ จุดยอด ลูกศรแต่ละเส้นเชื่อมต่อดูดยอดไปยังพ่อของจุดนั้น ยกเว้นราก ซึ่งไม่มีพ่อ ต้นไม้ย่อยของจุดยอด 2 ประกอบด้วยจุดยอด 2, 3, 4 และ 5 ต้นไม้ย่อยของจุดยอด 0 ประกอบด้วยจุดยอดทั้งหมด 6 จุดยอดของต้นไม้ และต้นไม้ย่อยของจุดยอด 4 ก็ประกอบด้วยจุดยอด 4 เท่านั้น



กำหนด **น้ำหนัก** ให้จุดยอดแต่ละจุด เป็นจำนวนเต็มไม่ติดลบ เราเขียนระบุน้ำหนักของจุดยอด i ($0 \leq i < N$) ว่า $W[i]$

หน้าที่ของคุณคือเขียนโปรแกรมที่จะตอบคำถาม Q คำถาม คำถามแต่ละข้อเขียนระบุด้วยจำนวนเต็มบวกสองตัว (L, R) โดยการตอบคำถามนั้นให้คำนวณดังนี้

พิจารณาการกำหนดค่าจำนวนเต็ม ซึ่งเรียกว่า **สัมประสิทธิ์** ให้กับแต่ละจุดยอดของต้นไม้ การกำหนดค่าดังกล่าวทำได้โดยเขียนลำดับ $C[0], \dots, C[N - 1]$ โดยที่ $C[i]$ ($0 \leq i < N$) คือค่าสัมประสิทธิ์ที่เรากำหนดให้กับจุดยอด i ขอให้เรียกลำดับนี้ว่า **ลำดับสัมประสิทธิ์** สังเกตว่าสมาชิกของลำดับสัมประสิทธิ์นั้นอาจติดลบ เป็น 0 หรือเป็นบวกก็ได้

สำหรับคำถาม (L, R) เราจะถือว่าลำดับสัมประสิทธิ์ **ขอบรรลุ** ถ้า สำหรับทุกๆ จุดยอด i ($0 \leq i < N$) เซตต่อไปนี้ เป็นจริง: ผลรวมสัมประสิทธิ์ของจุดต่างๆ ในต้นไม้ย่อยของจุดยอด i มีค่าไม่น้อยกว่า L และไม่มากกว่า R

สำหรับลำดับสัมประสิทธิ์ $C[0], \dots, C[N - 1]$ **ราคา** ของจุดยอด i คือ $|C[i]| \cdot W[i]$ โดยที่ $|C[i]|$ หมายถึงค่าสัมบูรณ์ของ $C[i]$ ในที่สุดแล้ว **ราคารวม** ก็คือผลรวมของราคาของจุดยอดทุกจุด หน้าที่ของคุณในการตอบคำถาม

แต่ละข้อคือการคำนวณหา **ราคารวมที่ต่ำที่สุด** ที่เป็นไปได้จากลำดับสัมประสิทธิ์ที่ขอบรรรม
สามารถพิสูจน์ได้ว่าสำหรับคำถามทุกข้อ มีลำดับสัมประสิทธิ์ที่ขอบรรรมอย่างน้อยหนึ่งลำดับ

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันสองฟังก์ชันต่อไปนี้:

```
void init(std::vector<int> P, std::vector<int> W)
```

- P, W : อาร์เรย์จำนวนเต็ม ขนาด N กำหนดพ่อ และน้ำหนัก ของแต่ละจุดยอด
- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้งเท่านั้น ตอนเริ่มต้นการคุยกันระหว่างเกรดเดอร์กับโปรแกรมของคุณในแต่ละกรณีทดสอบ

```
long long query(int L, int R)
```

- L, R : จำนวนเต็มที่ระบุคำถาม
- ฟังก์ชันนี้จะถูกเรียก Q ครั้ง หลังจากที่เราเรียก `init` ในแต่ละกรณีทดสอบ
- ฟังก์ชันนี้ควรคืนค่าคำตอบ สำหรับคำถามที่กำหนดให้

เงื่อนไข

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ สำหรับแต่ละ i โดยที่ $1 \leq i < N$
- $0 \leq W[i] \leq 1\,000\,000$ สำหรับแต่ละ i โดยที่ $0 \leq i < N$
- $1 \leq L \leq R \leq 1\,000\,000$ ในแต่ละคำถาม

ปัญหาย่อย

ปัญหาย่อย	คะแนน	เงื่อนไขเพิ่มเติม
1	10	$Q \leq 10; W[P[i]] \leq W[i]$ สำหรับแต่ละ i โดยที่ $1 \leq i < N$
2	13	$Q \leq 10; N \leq 2\,000$
3	18	$Q \leq 10; N \leq 60\,000$
4	7	$W[i] = 1$ สำหรับแต่ละ i โดยที่ $0 \leq i < N$
5	11	$W[i] \leq 1$ สำหรับแต่ละ i โดยที่ $0 \leq i < N$
6	22	$L = 1$
7	19	ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

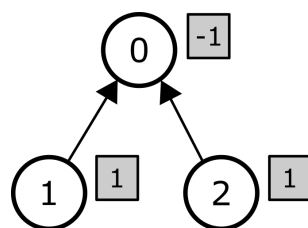
พิจารณาการเรียกฟังก์ชันนี้:

```
init([-1, 0, 0], [1, 1, 1])
```

ต้นไม้มี 3 จุดยอด นั่นคือ รากและลูกของราก 2 จุดยอด ทุกจุดยอดมีน้ำหนัก 1

```
query(1, 1)
```

ในคำถามนี้ $L = R = 1$ ซึ่งหมายความว่า ผลรวมของสัมประสิทธิ์ของต้นไม้ย่อยทุกต้น จะต้องเท่ากับ 1 พิจารณาลำดับสัมประสิทธิ์ $[-1, 1, 1]$ ต้นไม้และสัมประสิทธิ์ที่สอดคล้อง (ในสี่เหลี่ยมเทา) แสดงอยู่ในภาพด้านล่าง



สำหรับทุกๆ จุดยอด i ($0 \leq i < 3$) ผลรวมสัมประสิทธิ์ของทุกจุดยอดในต้นไม้ย่อยของ i เท่ากับ 1 ดังนั้น ถือว่าลำดับสัมประสิทธิ์นี้ชอบธรรม คำนวณราคารวมได้ดังนี้:

จุดยอด	น้ำหนัก	สัมประสิทธิ์	ราคา
0	1	-1	$ -1 \cdot 1 = 1$
1	1	1	$ 1 \cdot 1 = 1$
2	1	1	$ 1 \cdot 1 = 1$

ดังนั้น ราคารวมคือ 3 ลำดับสัมประสิทธิ์นี้เป็นลำดับเดียวที่ชอบธรรม ดังนั้น ฟังก์ชันนี้ควรคืนค่า 3

```
query(1, 2)
```

ราคารวมที่ต่ำที่สุดของคำถามนี้คือ 2 และได้มาจากลำดับสัมประสิทธิ์ $[0, 1, 1]$

เกร็ดเดอร์ตัวอย่าง

รูปแบบข้อมูลนำเข้า:

```
N
P[1]  P[2]  ...  P[N-1]
W[0]  W[1]  ...  W[N-2] W[N-1]
Q
L[0]  R[0]
L[1]  R[1]
...
L[Q-1] R[Q-1]
```

โดยที่ $L[j]$ และ $R[j]$ (สำหรับ $0 \leq j < Q$) คืออาร์กิวเมนต์สำหรับการเรียก query ครั้งที่ j สังเกตว่าบรรทัดที่สองของข้อมูลนำเข้า มีจำนวนเต็มเพียง $N - 1$ ตัวเท่านั้น เพราะเรดเดอร์ตัวอย่างไม่อ่านค่าของ $P[0]$

รูปแบบข้อมูลส่งออก:

```
A[0]
A[1]
...
A[Q-1]
```

โดยที่ $A[j]$ (สำหรับ $0 \leq j < Q$) คือค่าที่ได้รับคืนจากการเรียก query ครั้งที่ j