

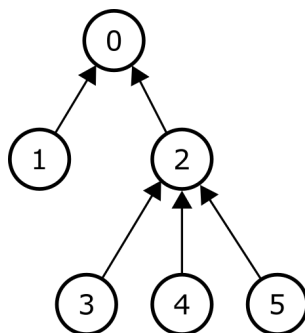
Дерево

Дерево складається з N **вершин**, пронумерованих від 0 до $N - 1$. Вершина 0 називається **коренем**. Кожна вершина, окрім кореня, має рівно одного **батька**. Для кожного i такого, що $1 \leq i < N$, батько вершини i - це вершина $P[i]$, де $P[i] < i$. Припустимо, що $P[0] = -1$.

Для кожної вершини i ($0 \leq i < N$), **піддерево** вершини i - це множина наступних вершин:

- i та
- всі вершини, чий батько i , та
- всі вершини, чий батько батька i , та
- всі вершини, чий батько батька батька i , та
- і так далі.

Малюнок нижче показує приклад дерева, що складається з $N = 6$ вершин. Кожна стрілка з'єднує вершину з її батьком, крім кореня, який не має батька. Піддерево вершини 2 містить вершини 2, 3, 4 і 5. Піддерево вершини 0 містить всі 6 вершин дерева і піддерево вершини 4 містить лише вершину 4.



Кожна вершина має невід'ємну цілу **вагу**. Ми позначимо $W[i]$ як вагу вершини i ($0 \leq i < N$).

Ваше завдання написати програму, яка відповідає на Q запитів, кожен з яких описується парою цілих додатніх чисел (L, R) . Відповідь на кожен запит має рахуватися наступним чином.

Кожній вершині дерева ми назначимо цілочисельний **коефіцієнт**. Вони описуються послідовністю $C[0], \dots, C[N - 1]$, де $C[i]$ ($0 \leq i < N$) - це коефіцієнт, який назначений вершині i . Назвемо цю послідовність **послідовністю коефіцієнтів**. Зверніть увагу, що всі числа цієї послідовності коефіцієнтів можуть бути від'ємними, 0, або додатніми.

Для запиту (L, R) , послідовність коефіцієнтів називається **правильною**, якщо для кожної вершини i ($0 \leq i < N$), виконується така умова: сума коефіцієнтів вершин у піддереві вершини i не менше L і не більше R .

Для певної послідовності коефіцієнтів $C[0], \dots, C[N-1]$, **вартість** вершини i дорівнює $|C[i]| \cdot W[i]$, де $|C[i]|$ позначає абсолютне значення $C[i]$. Нарешті, **загальна вартість** є сумою вартостей усіх вершин. Ваше завдання полягає в тому, щоб обчислити для кожного запиту **мінімальну загальну вартість**, яка може бути досягнута певною правильною послідовністю коефіцієнтів.

Можна показати, що для будь-якого запиту існує принаймні одна правильна послідовність коефіцієнтів.

Деталі реалізації

Ви повинні реалізувати наступні дві функції:

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

- P, W : масиви цілих чисел довжини N із зазначенням батьків і ваги.
- Ця функція викликається рівно один раз на початку взаємодії між градером і вашою програмою в кожному тестовому випадку.

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

- L, R : цілі числа, що описують запит.
- Ця функція викликається Q разів після виклику `init` у кожному тестовому випадку.
- Ця функція має повернути відповідь на заданий запит.

Обмеження

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ для кожного i такого, що $1 \leq i < N$
- $0 \leq W[i] \leq 1\,000\,000$ для кожного i такого, що $0 \leq i < N$
- $1 \leq L \leq R \leq 1\,000\,000$ у кожному запиті

Підзадачі

Підзадача	Балів	Додаткові обмеження
1	10	$Q \leq 10; W[P[i]] \leq W[i]$ для кожного i такого, що $1 \leq i < N$
2	13	$Q \leq 10; N \leq 2\,000$
3	18	$Q \leq 10; N \leq 60\,000$
4	7	$W[i] = 1$ для кожного i такого, що $0 \leq i < N$
5	11	$W[i] \leq 1$ для кожного i такого, що $0 \leq i < N$
6	22	$L = 1$
7	19	Без додаткових обмежень.

Приклад

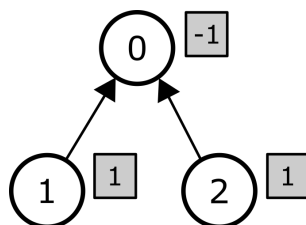
Розглянемо наступні виклики:

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

Дерево складається з 3 вершин: кореня та його 2 дітей. Усі вершини мають вагу 1.

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

У цьому запиті $L = R = 1$, це означає, що сума коефіцієнтів у кожному піддереві має дорівнювати 1. Розглянемо послідовність коефіцієнтів $[-1, 1, 1]$. Нижче наведено дерево та відповідні коефіцієнти (у заштрихованих прямокутниках).



Для кожної вершини i ($0 \leq i < 3$) сума коефіцієнтів усіх вершин у піддереві i дорівнює 1. Отже, ця послідовність коефіцієнтів правильна. Загальна вартість розраховується наступним чином:

Вершина	Вага	Коефіцієнт	Вартість
0	1	-1	$ -1 \cdot 1 = 1$
1	1	1	$ 1 \cdot 1 = 1$
2	1	1	$ 1 \cdot 1 = 1$

Тому загальна вартість становить 3. Це єдина правильна послідовність коефіцієнтів, тому цей виклик має повернути 3.

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

Мінімальна загальна вартість цього запиту становить 2, і досягається, коли послідовність коефіцієнтів дорівнює $[0, 1, 1]$.

Приклад градера

Формат вхідних даних:

```
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]
```

де $L[j]$ і $R[j]$ (для $0 \leq j < Q$) є вхідними аргументами в j -му виклику query. Зверніть увагу, що другий рядок містить **тільки** $N - 1$ **цілих чисел**, оскільки приклад градера не зчитує значення $P[0]$.

Формат вихідних даних:

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

де $A[j]$ (для $0 \leq j < Q$) - це значення, яке повертає j -й виклик query.