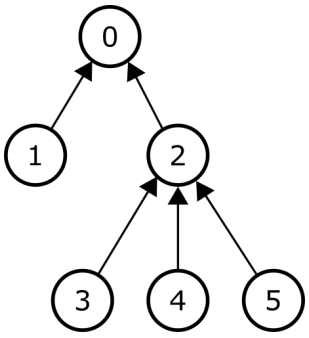
Boom

Neem een **boom** bestaande uit N **knopen**, genummerd van 0 tot en met $N-1$. Knoop 0 wordt de **wortel** genoemd. Elke knoop, behalve de wortel, heeft één **ouder**. Voor elke i ($1 \leq i < N$) is de ouder van knoop i knoop $P[i]$, waar $P[i] < i$. We nemen ook aan dat $P[0] = -1$.

Voor elke knoop i ($0 \leq i < N$), de **subboom** van i is de verzameling van de volgende knopen:

- i, en
- elke knoop waarvan de ouder i is, en
- elke knoop waarvan de ouder van de ouder i is, en
- elke knoop waarvan de ouder van de ouder van de ouder i is, en
- enz.

De onderstaande afbeelding toont een voorbeeldboom bestaande uit $N = 6$ knopen. Elke pijl verbindt een knoop met zijn ouder, behalve de wortel, die geen ouder heeft. De subboom van knoop 2 bevat knopen $2, 3, 4$ en 5. De subboom van knoop 0 bevat alle 6 de knopen van de boom en de subboom van knoop 4 bevat alleen knoop 4.



Aan elke knoop wordt een niet-negatief integer **gewicht** gegeven. Het gewicht van knoop i ($0 \leq i < N$) is gegeven door $W[i]$.

Jouw taak is om een programma te schrijven dat Q queries beantwoordt, elk bestaand uit twee positieve integers (L, R). Het antwoord op de query moet als volgt worden berekend.

Geef elke knoop in de boom een integer waarde, de **coëfficiënt**. Deze toewijzing wordt beschreven door de reeks $C[0], \ldots, C[N-1]$, waarbij $C[i]$ ($0 \leq i < N$) de coëfficiënt is die aan knoop i is gegeven. Deze reeks noemen we de **coëfficiëntenreeks**. Let erop dat de elementen van de coëfficiëntenreeks negatief, 0 en positief kunnen zijn.

Voor een query (L, R) wordt een coëfficiëntenreeks **geldig** genoemd als voor elke knoop i ($0 \leq i < N$) geldt: de som van de coëfficiënten van de knopen in de subboom van knoop i is niet kleiner dan L en niet groter dan R.

Voor een gegeven coëfficiëntenreeks $C[0], \ldots, C[N-1]$, zijn de **kosten** van een knoop i $|C[i]| \cdot W[i]$, waarbij $|C[i]|$ de absolute waarde van $C[i]$ aangeeft. De **totale kosten** zijn de som van de kosten van alle knopen. Jouw taak is om voor elke query te berekenen: de **minimale totale kosten** die je kan krijgen met een geldige coëfficiëntenreeks.

Je kan bewijzen dat voor elke mogelijke query, er ten minste 1 geldige coëfficiëntenreeks bestaat.

Implementatie Details

Je moet de volgende twee procedures implementeren:

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

- P, W: arrays van integers met lengte N die de ouders en de gewichten beschrijven.
- Deze procedure wordt exact één keer aangeroepen In het begin van de interactie tussen de grader en jouw programma in elke test case.

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

- L, R: integers die een query beschrijven.
- Deze procedure wordt Q keer aangeroepen na `init` in elke test case.
- Deze procedure moet het antwoord van de gegeven query returnen.

Beperkingen

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ voor elke i zodat $1 \leq i < N$
- $0 \leq W[i] \leq 1\,000\,000$ voor elke i zodat $0 \leq i < N$
- $1 \leq L \leq R \leq 1\,000\,000$ in elke query

Subtaken

Subtaak	Score	Extra beperkingen
1	10	$Q \leq 10$; $W[P[i]] \leq W[i]$ voor elke i zodat $1 \leq i < N$
2	13	$Q \leq 10$; $N \leq 2\,000$
3	18	$Q \leq 10$; $N \leq 60\,000$
4	7	$W[i] = 1$ voor elke i zodat $0 \leq i < N$
5	11	$W[i] \leq 1$ voor elke i zodat $0 \leq i < N$
6	22	$L = 1$
7	19	Geen extra beperkingen.

Voorbeelden

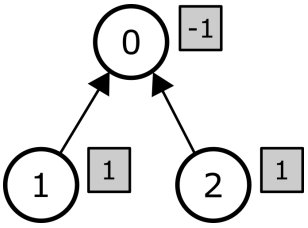
Bekijk de volgende calls.

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

De boom bestaat uit 3 knopen, de wortel en zijn 2 kinderen. Alle knopen hebben gewicht 1.

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

In deze query $L = R = 1$, wat betekent dat de som van de coëfficiënten in elke subboom gelijk moet zijn aan 1. Neem bijvoorbeeld de coëfficiëntenreeks $[-1, 1, 1]$. De boom en de bijbehorende coëfficiënten (in getinte rechthoeken) is hieronder getekend.



Voor elke knoop i ($0 \leq i < 3$) is de som van de coëfficiënten van alle knopen in de subboom van i gelijk aan 1. Deze coëfficiëntenreeks is dus geldig. De totale kosten worden als volgt berekend:

Knoop	Gewicht	Coëfficiënt	Kosten
0	1	-1	$\lvert -1 \rvert \cdot 1 = 1$
1	1	1	$\lvert 1 \rvert \cdot 1 = 1$
2	1	1	$\lvert 1 \rvert \cdot 1 = 1$

De totale kosten zijn dus 3. Dit is de enige geldige coëfficiëntenreeks, daarom zou deze call 3 moeten returnen.

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

De minimale totale kosten voor deze query zijn 2. Deze kosten krijg je met de coëfficiëntenreeks $[0, 1, 1]$.

Voorbeeldgrader

Invoerformaat:

```
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]
```

waar $L[j]$ en $R[j]$ (voor $0 \leq j < Q$) de invoerargumenten zijn in de j-de aanroep van `query`. Let erop dat de tweede regel van de invoer **alleen** $N - 1$ **integers** bevat, omdat de voorbeeldgrader de waarde van $P[0]$ niet leest.

Uitvoerformaat:

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

waar $A[j]$ (voor $0 \leq j < Q$) de waarde is die wordt gereturnt door de j-de aanroep van `query`.