

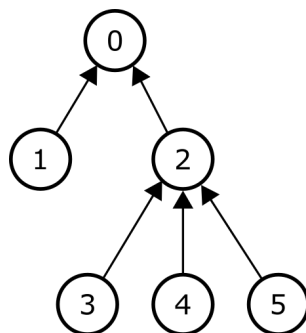
Drevo

Obravnavajmo **drevo**, sestavljeno iz N **vozlišč**, oštevilčenih od 0 do $N - 1$. Vozlišče 0 se imenuje **koren**. Vsako vozlišče, razen korena, ima enega samega **starša**. Za vsak i , ki je tak, da $1 \leq i < N$, je $P[i]$ starš vozlišča i , kjer $P[i] < i$. Predpostavljamo tudi, da $P[0] = -1$.

Za vsako vozlišče i ($0 \leq i < N$), je **poddrevo** i množica naslednjih vozlišč:

- i in
- vsa vozlišča, katerim je i starš, in
- vsa vozlišča, katerim staršev starš je i , in
- vsa vozlišča, katerih staršev staršev starš je i , in
- tako naprej.

Spodnja slika prikazuje primer drevesa, sestavljenega iz $N = 6$ vozlišč. Vsaka puščica povezuje vozlišče s svojim staršem, razen korena, ki nima starša. Poddrevo vozlišča 2 vsebuje vozlišča 2, 3, 4 in 5. Poddrevo vozlišča 0 vsebuje vseh 6 vozlišč drevesa in poddrevo vozlišča 4 vsebuje zgolj vozlišče 4.



Vsakemu vozlišču je dodeljena nenegativna celoštevilska **utež**. Utež vozlišča i ($0 \leq i < N$) označimo z $W[i]$.

Vaša naloga je napisati program, ki bo odgovoril na Q poizvedb, kjer je vsako vprašanje podano s pozitivnima celima številoma (L, R) . Odgovor na poizvedbo je treba izračunati na naslednji način.

Vsakemu vozlišču drevesa dodelimo celo število, ki imenujemo ga **koeficient**. Ta dodelitev je opisana z zaporedjem $C[0], \dots, C[N - 1]$, kjer je $C[i]$ ($0 \leq i < N$) koeficient, dodeljen vozlišču i . To zaporedje imenujmo **zaporedje koeficientov**. Upoštevajte, da so elementi zaporedja koeficientov lahko negativni, 0 ali pozitivni.

Za poizvedbo (L, R) , se zaporedje koeficientov imenuje **veljavno**, če za vsako vozlišče i ($0 \leq i < N$), velja naslednji pogoj: vsota koeficientov vozlišč v poddrevesu vozlišča i ni manjša od L in ni večja od R .

Za dano zaporedje koeficientov $C[0], \dots, C[N-1]$, je **cena** vozlišča i $|C[i]| \cdot W[i]$, kjer $|C[i]|$ označuje absolutno vrednost $C[i]$. **Skupna cena** je vsota cen vseh vozlišč. Vaša naloga je za vsako poizvedbo izračunati **najmanjšo skupno ceno**, ki jo je mogoče doseči z nekim veljavnim zaporedjem koeficientov.

Lahko bi pokazali, da za katero koli poizvedbo obstaja vsaj eno veljavno zaporedje koeficientov.

Podrobnosti implementacije

Implementirajte naslednjo proceduro:

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

- P, W : polji celih števil dolžine N , navedba staršev in uteži.
- Proceduro se kliče natanko enkrat, na začetku interakcije med ocenjevalnikom in vašim programom, pri vsakem testnem primeru.

In naslednjo funkcijo:

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

- L, R : celi števili, ki opisujeta poizvedbo.
- Funkcijo se kliče Q -krat, po priklicu `init`, pri vsakem testnem primeru.
- Funkcija naj vrne odgovor na podano poizvedbo.

Omejitve

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ za vsak i velja $1 \leq i < N$
- $0 \leq W[i] \leq 1\,000\,000$ za vsak i velja $0 \leq i < N$
- $1 \leq L \leq R \leq 1\,000\,000$, pri vsaki poizvedbi

Podnaloge

Podnaloga	Točke	Dodatne omejitve
1	10	$Q \leq 10$; $W[P[i]] \leq W[i]$ za vsak i velja $1 \leq i < N$
2	13	$Q \leq 10$; $N \leq 2\,000$
3	18	$Q \leq 10$; $N \leq 60\,000$
4	7	$W[i] = 1$ za vsak i velja $0 \leq i < N$
5	11	$W[i] \leq 1$ za vsak i velja $0 \leq i < N$
6	22	$L = 1$
7	19	Brez dodatnih omejitev.

Primer

Razmislite o naslednjih klicih:

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

Drevo je sestavljeno iz 3 vozlišč, korena in njegovih 2 otrok. Vsa vozlišča imajo težo 1.

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

Pri tej poizvedbi velja $L = R = 1$, kar pomeni, da mora biti vsota koeficientov v vsakem poddrevesu enaka 1. Razmislite o zaporedju koeficientov $[-1, 1, 1]$. Drevo in ustrezni koeficienti (v osenčenih pravokotnikih) so prikazani spodaj.



Za vsako vozlišče i ($0 \leq i < 3$) je vsota koeficientov vseh vozlišč poddrevesa vozlišča i enaka 1. Zato je to zaporedje koeficientov veljavno. Skupna cena se izračuna na naslednji način:

Vozlišče	Utež	Koeficient	Cena
0	1	-1	$ -1 \cdot 1 = 1$
1	1	1	$ 1 \cdot 1 = 1$
2	1	1	$ 1 \cdot 1 = 1$

Zatorej je skupna cena 3. To je edino veljavno zaporedje koeficientov, zato bi ta klic moral vrniti 3.

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

Najmanjša skupna cena za to poizvedbo je 2, in je dosežena, ko je zaporedje koeficientov $[0, 1, 1]$.

Vzorčni ocenjevalnik

Oblika vhoda:

```
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]
```

kjer sta $L[j]$ in $R[j]$ (za $0 \leq j < Q$) vhodna argumenta pri j -tem klicu query. Upoštevajte, da druga vrstica vnosa vsebuje **samo** $N - 1$ celih števil, saj vzorčni ocenjevalnik ne prebere vrednosti $P[0]$.

Oblika izhoda:

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

kjer je $A[j]$ (za $0 \leq j < Q$) vrednost, ki jo vrne j -ti klic query.