

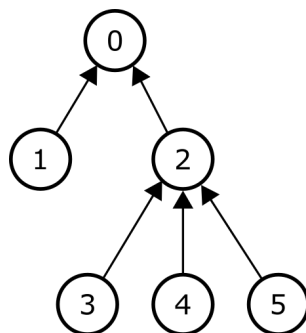
Strom

Uvažme **strom** tvořený N **vrcholy** očíslovanými 0 až $N - 1$. Vrchol 0 nazýváme **kořen**. Každý vrchol, kromě kořene, má jednoho **rodiče**. Pro každé i ($1 \leq i \leq N$) je rodič i vrchol $P[i]$, kde $P[i] \leq i$. Dále předpokládáme, že $P[0] = -1$.

Pro každý vrchol i je **podstrom** i množina těchto vrcholů:

- vrchol i ,
- jakýkoliv vrchol, jehož rodič je i ,
- jakýkoliv vrchol, jehož prarodič je i ,
- jakýkoliv vrchol, jehož praprarodič je i ,
- a tak dále...

Obrázek níže ukazuje příklad stromu z $N = 6$ vrcholů. Každá šipka vede od vrcholu k jeho rodiči, kromě kořene, který žádného rodiče nemá. Podstrom vrcholu 2 obsahuje vrcholy $2, 3, 4$ a 5 . Podstrom vrcholu 0 obsahuje všech 6 vrcholů stromu a podstrom vrcholu 4 obsahuje pouze vrchol 4 .



Každému vrcholu je přiřazena nezáporná celočíselná **váha**. Váhu vrcholu i ($0 \leq i < N$) značíme $W[i]$.

Vášim úkolem je napsat program, který zodpoví Q dotazů, každý určený dvojicí kladných celých čísel (L, R) . Odpověď na dotaz je spočítána následovně:

Přiřadíme každému vrcholu celé číslo, nazývané **koeficient**. Přiřazení je popsáno posloupností $C[0], \dots, C[N - 1]$, kde $C[i]$ ($0 \leq i < N$) je koeficient přiřazený vrcholu i . Toto přiřazení nazýváme **posloupnost koeficientů**. Dodejme, že prvky posloupnosti koeficientů mohou být záporné, 0 , nebo kladné.

Pro dotaz (L, R) je posloupnost koeficientů **validní**, pokud pro každý vrchol i ($0 \leq i < N$) platí následující podmínka: Součet koeficientů vrcholů v podstromu vrcholu i není menší než L ani větší než R .

Pro danou posloupnost koeficientů $C[0], \dots, C[N-1]$, **cena** vrcholu i je $|C[i]| \cdot W[i]$, kde $|C[i]|$ značí absolutní hodnotu $C[i]$. **Celková cena** je součet cen všech vrcholů. Vaším úkolem je pro každý dotaz spočítat **minimální celkovou cenu**, kterou jde získat validní posloupností koeficientů.

Lze dokázat, že pro každý dotaz existuje alespoň jedna validní posloupnost koeficientů.

Implementační detaily

Máte za úkol implementovat následující dvě funkce:

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

- P, W : pole celých čísel délky N specifikující rodiče a váhy.
- Tato funkce je volána pouze jednou na začátku interakce mezi graderem a vaším programem na každém vstupu.

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

- L, R : celá čísla popisující dotaz.
- Tato funkce bude zavolána Q -krát po zavolání `init` na každém vstupu.
- Tato funkce by měla vrátit odpověď na zadaný dotaz.

Omezení

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ pro každé i , kde $1 \leq i < N$
- $0 \leq W[i] \leq 1\,000\,000$ pro každé i , kde $0 \leq i < N$
- $1 \leq L \leq R \leq 1\,000\,000$ v každém dotazu

Podúlohy

Podúloha	Počet bodů	Dodatečná omezení
1	10	$Q \leq 10; W[P[i]] \leq W[i]$ pro každé i takové, že $1 \leq i < N$
2	13	$Q \leq 10; N \leq 2\,000$
3	18	$Q \leq 10; N \leq 60\,000$
4	7	$W[i] = 1$ pro každé i takové, že $0 \leq i < N$
5	11	$W[i] \leq 1$ pro každé i takové, že $0 \leq i < N$
6	22	$L = 1$
7	19	Bez dalších omezení.

Příklady

Uvažme následující zavolání:

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

Strom se skládá ze 3 vrcholů, kořene se dvěma dětmi. Všechny vrcholy mají váhu 1.

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

V tomto dotazu $L = R = 1$, což znamená, že součet koeficientů v každém podstromu musí být 1. Uvažme posloupnost koeficientů $[-1, 1, 1]$. Strom a odpovídající koeficienty (v šedých obdélnících) jsou vyobrazeny níže:



Pro každý vrchol i ($0 \leq i < 3$), součet koeficientů všech vrcholů podstromu i je 1. Tedy tato posloupnost koeficientů je validní. Celková cena je spočtena následovně:

Vrchol	Váha	Koeficient	Cena
0	1	-1	$ -1 \cdot 1 = 1$
1	1	1	$ 1 \cdot 1 = 1$
2	1	1	$ 1 \cdot 1 = 1$

Tedy celková cena je 3. Toto je jediná validní posloupnost koeficientů, a proto toto zavolání má vrátit 3.

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

Minimální celková cena pro tento dotaz je 2, a je dosažena posloupností koeficientů $[0, 1, 1]$.

Ukázkový grader

Formát vstupu:

```
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]
```

kde $L[j]$ a $R[j]$ (pro $0 \leq j < Q$) jsou vstupní argumenty v j -tém zavolání query. Poznamenejme, že druhý řádek vstupu obsahuje **pouze** $N - 1$ čísel, protože ukázkový grader nečte hodnotu $P[0]$.

Note that the second line of the input contains **only** $N - 1$ integers, as the sample grader does not read the value of $P[0]$.

Formát výstupu:

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

kde $A[j]$ (pro $0 \leq j < Q$) je hodnota vrácená j -tým zavoláním query.