

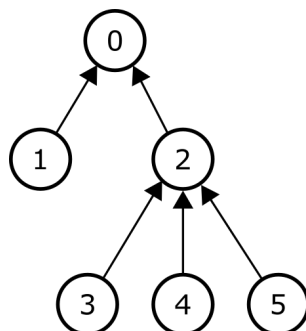
# Tree

Consideriamo un **albero** costituito da  $N$  **nodi**, numerati da 0 a  $N - 1$ . Il nodo 0 è chiamato **radice**. Ogni nodo, eccetto la radice, ha un singolo **parent**. Per ogni  $1 \leq i < N$ , il parent del nodo  $i$  è il nodo  $P[i]$  con  $P[i] < i$ , e per convenzione  $P[0] = -1$ .

Per ogni nodo  $i$  ( $0 \leq i < N$ ), il **sottoalbero** di  $i$  è l'insieme dei seguenti nodi:

- $i$ , e
- qualsiasi nodo il cui parent è  $i$ , e
- qualsiasi nodo il cui parent del parent è  $i$ , e
- qualsiasi nodo il cui parent del parent del parent è  $i$ , e
- ecc...

L'immagine seguente mostra un albero di esempio costituito da  $N = 6$  nodi. Ogni freccia collega un nodo al suo parent, ad eccezione della radice, che non ha alcun parent. Il sottoalbero del nodo 2 contiene i nodi 2, 3, 4 e 5. Il sottoalbero del nodo 0 contiene tutti e 6 i nodi dell'albero e il sottoalbero del nodo 4 contiene solo il nodo 4.



A ciascun nodo  $0 \leq i < N$  è assegnato un **peso**  $W[i]$  intero non negativo.

Devi scrivere un programma che risponda a  $Q$  query, ciascuna specificata da una coppia di interi positivi  $(L, R)$ . La risposta alla query deve essere calcolata come segue.

Considera l'assegnazione di un numero intero  $C[i]$  chiamato **coefficiente** a ciascun nodo  $i$  dell'albero ( $0 \leq i < N$ ). Chiamiamo la sequenza  $C[0], \dots, C[N - 1]$  **sequenza di coefficienti**. Si noti che gli elementi della sequenza dei coefficienti possono essere negativi, 0 o positivi.

Per una query  $(L, R)$ , una sequenza di coefficienti è detta **valida** se, per ogni nodo  $i$  ( $0 \leq i < N$ ), vale la seguente condizione: la somma dei coefficienti dei nodi nel sottoalbero del nodo  $i$  non è minore di  $L$  e non è maggiore di  $R$ .

Data una sequenza di coefficienti  $C[0], \dots, C[N-1]$ , il **costo** di un nodo  $i$  è  $|C[i]| \cdot W[i]$  (dove  $|C[i]|$  denota il valore assoluto di  $C[i]$ ) e il **costo totale** è la somma dei costi di tutti i nodi. Il tuo compito è calcolare, per ogni query, il **costo totale minimo** che può avere una sequenza di coefficienti valida.

È possibile dimostrare che per qualsiasi query esiste almeno una sequenza di coefficienti valida.

## Note di implementazione

Devi implementare le seguenti funzioni:

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

- $P, W$ : array di interi di lunghezza  $N$ , contenenti rispettivamente i parent e i pesi.
- Questa funzione viene chiamata esattamente una volta all'inizio dell'esecuzione di ogni caso di test.

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

- $L, R$ : numeri interi che descrivono una query.
- Questa funzione viene chiamata  $Q$  volte dopo l'invocazione di `init` in ogni caso di test.
- Questa funzione deve restituire la risposta alla query specificata.

## Assunzioni

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$  per ogni  $i$  tale che  $1 \leq i < N$
- $0 \leq W[i] \leq 1\,000\,000$  per ogni  $i$  tale che  $0 \leq i < N$
- $1 \leq L \leq R \leq 1\,000\,000$  in ogni query

## Subtask

Subtask	Punteggio	Limitazioni aggiuntive
1	10	$Q \leq 10; W[P[i]] \leq W[i]$ per ogni $1 \leq i < N$
2	13	$Q \leq 10; N \leq 2\,000$
3	18	$Q \leq 10; N \leq 60\,000$
4	7	$W[i] = 1$ per ogni $0 \leq i < N$
5	11	$W[i] \leq 1$ per ogni $0 \leq i < N$
6	22	$L = 1$
7	19	Nessuna limitazione aggiuntiva.

## Esempio

Consideriamo le seguenti chiamate:

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

L'albero è composto da 3 nodi: la radice e i suoi 2 figli. Tutti i nodi hanno peso 1.

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

In questa query  $L = R = 1$ , il che significa che la somma dei coefficienti in ogni sottoalbero deve essere uguale a 1. Consideriamo la sequenza di coefficienti  $[-1, 1, 1]$ . Di seguito sono illustrati l'albero e i coefficienti corrispondenti (nei rettangoli ombreggiati).



Per ogni nodo  $i$  ( $0 \leq i < 3$ ), la somma dei coefficienti di tutti i nodi nel sottoalbero di  $i$  è uguale a 1, quindi questa sequenza di coefficienti è valida. Il costo totale è calcolato come segue:

Nodo	Peso	Coefficiente	Costo
0	1	-1	$ -1  \cdot 1 = 1$
1	1	1	$ 1  \cdot 1 = 1$
2	1	1	$ 1  \cdot 1 = 1$

Quindi il costo totale è 3. Questa è l'unica sequenza di coefficienti valida, quindi questa chiamata deve restituire 3.

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

Il costo totale minimo per quest'altra query è 2, e si ottiene quando la sequenza dei coefficienti è  $[0, 1, 1]$ .

## Grader di esempio

Formato di input:

```
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]
```

dove  $L[j]$  e  $R[j]$  (per  $0 \leq j < Q$ ) sono gli argomenti di input nella  $j$ -esima chiamata a query. Si noti che la seconda riga dell'input contiene **solo**  $N - 1$  **interi**, poiché il grader di esempio non legge il valore di  $P[0]$ .

Formato di output:

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

dove  $A[j]$  (per  $0 \leq j < Q$ ) è il valore restituito dalla  $j$ -esima chiamata a query.