

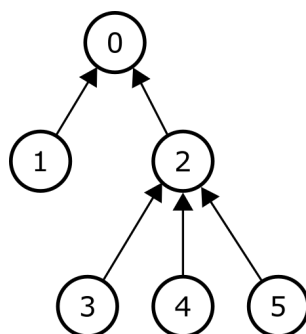
Arbre (Tree)

Considérons un **arbre** constitué de N **sommets**, numérotés de 0 à $N - 1$. Le sommet 0 est appelé la **racine**. Chaque sommet, à l'exception de la racine, a un unique **parent**. Pour chaque i tel que $1 \leq i < N$, le parent du sommet i est le sommet $P[i]$, où $P[i] < i$. On suppose aussi que $P[0] = -1$.

Pour chaque sommet i ($0 \leq i < N$), le **sous-arbre** de i est l'ensemble qui contient les sommets suivants :

- i ,
- les sommets dont le parent est i ,
- les sommets dont le parent du parent est i ,
- les sommets dont le parent du parent du parent est i ,
- ...

La figure ci-dessous montre un arbre qui est constitué de $N = 6$ sommets. Chaque flèche connecte un sommet à son parent, sauf pour la racine qui n'a pas de parent. Le sous-arbre du sommet 2 contient les sommets $2, 3, 4$ et 5 . Le sous-arbre du sommet 0 contient les 6 sommets de l'arbre et le sous-arbre du sommet 4 contient uniquement le sommet 4 .



À chaque sommet, on associe un entier positif appelé **poids**. On utilisera la notation $W[i]$ pour désigner le poids du sommet i ($0 \leq i < N$).

Votre objectif est d'écrire un programme qui va répondre à Q requêtes. Chacune est décrite par une paire d'entiers strictement positifs (L, R) . La réponse à une requête doit être calculée de la manière suivante.

On associe à chaque sommet de l'arbre un entier appelé **coefficient**. Une telle association est décrite par une suite $C[0], \dots, C[N - 1]$, où $C[i]$ ($0 \leq i < N$) est le coefficient associé au sommet

i . Appelons cette suite la **séquence des coefficients**. Notez que les éléments de la séquence des coefficients peuvent être négatifs, 0, ou positifs.

Pour une requête (L, R) , une séquence de coefficients est **valide** si, pour chaque sommet i ($0 \leq i < N$), la condition suivante est vérifiée : la somme des coefficients des sommets du sous-arbre de i est supérieure ou égale à L et inférieure ou égale à R .

Pour une séquence de coefficients $C[0], \dots, C[N-1]$, le **coût** d'un sommet i est $|C[i]| \cdot W[i]$, où $|C[i]|$ désigne la valeur absolue de $C[i]$. Enfin, le **coût total** est la somme des coûts de tous les sommets. Votre tâche est de calculer, pour chaque requête, le **coût total minimum** qui peut être atteint par une séquence de coefficients valide.

Il peut être démontré que, pour chaque requête, il existe au moins une séquence de coefficients valide.

Détails d'implémentation

Vous devez implémenter les deux fonctions suivantes :

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

- Les tableaux P et W ont pour longueur N et définissent les parents et les poids.
- La fonction est appelée exactement une fois au début de l'interaction entre le grader et votre programme pour chaque test.

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

- Les entiers L et R décrivent une requête.
- Cette fonction est appelée Q fois après l'appel à `init` pour chaque test.
- Cette fonction doit renvoyer la réponse à la requête formulée.

Contraintes

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ pour chaque i tel que $1 \leq i < N$
- $0 \leq W[i] \leq 1\,000\,000$ pour chaque i tel que $0 \leq i < N$
- $1 \leq L \leq R \leq 1\,000\,000$ pour chaque requête

Sous-tâches

Sous-tâche	Score	Contraintes supplémentaires
1	10	$Q \leq 10$; $W[P[i]] \leq W[i]$ pour chaque i tel que $1 \leq i < N$
2	13	$Q \leq 10$; $N \leq 2\,000$
3	18	$Q \leq 10$; $N \leq 60\,000$
4	7	$W[i] = 1$ pour chaque i tel que $0 \leq i < N$
5	11	$W[i] \leq 1$ pour chaque i tel que $0 \leq i < N$
6	22	$L = 1$
7	19	Aucune contrainte supplémentaire.

Exemples

Considérons les appels suivants :

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

L'arbre est constitué de 3 sommets : la racine et ses 2 enfants. Tous les sommets ont un poids de 1.

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

Dans cette requête, on a $L = R = 1$. Cela signifie que la somme des coefficients de chaque sous-arbre doit être égale à 1. Considérons la séquence de coefficients $[-1, 1, 1]$. L'arbre et les coefficients correspondants (dans les rectangles grisés) sont illustrés ci-dessous.



Pour chaque sommet i ($0 \leq i < 3$), la somme des coefficients de tous les sommets dans le sous-arbre de i est égal à 1. Cette séquence de coefficients est donc valide. Le coût total est calculé de la manière suivante :

Sommet	Poids	Coefficient	Coût
0	1	-1	$ -1 \cdot 1 = 1$
1	1	1	$ 1 \cdot 1 = 1$
2	1	1	$ 1 \cdot 1 = 1$

Le coût total est donc de 3. Il s'agit de la seule séquence de coefficients valide, cet appel doit donc renvoyer 3.

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

Le coût minimum pour cette requête est 2, et est atteint par la séquence de coefficients $[0, 1, 1]$.

Évaluateur d'exemple (grader)

Format d'entrée :

```
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]
```

$L[j]$ et $R[j]$ (pour $0 \leq j < Q$) sont les arguments du j -ème appel à query. Notez que la deuxième ligne de l'entrée contient **seulement** $N - 1$ entiers, puisque le grader ne lit pas la valeur de $P[0]$.

Format de sortie :

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

$A[j]$ (pour $0 \leq j < Q$) est la valeur renvoyée par le j -ème appel à query.