

الشجرة

ليكن لدينا شجرة مكونة من N عقدة ، مرقمة من 0 إلى $N - 1$. العقدة 0 تدعى الجذر.

باقي العقد، باستثناء الجذر، لها أب واحد من أجل كل i ، حيث أن $1 \leq i < N$ ، الأب للعقدة i هو العقدة $P[i]$ ، حيث $P[i] < i$.

نفترض أن $P[0] = -1$.

لكل عقدة i ($0 \leq i < N$)، تكون الشجرة الفرعية للعقدة i هي مجموعة من العقد التالية:

- i ، و
- أي عقدة تكون i أباً لها، و
- أي عقدة تكون i أباً لأبيها (جداً لها)، و
- أي عقدة تكون i أباً لجدها، و
- وهكذا.

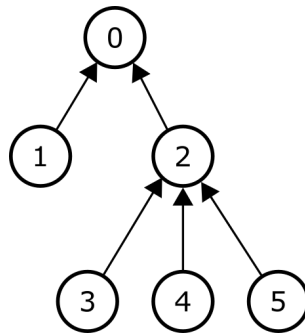
تظهر الصورة بالأسفل مثالاً لشجرة تحتوي على $N = 6$ عقدة.

كل سهم يصل عقدة بأبيها، ما عدا الجذر، والذي ليس له أب

الشجرة الفرعية للعقدة 2 تحتوي على العقد 3، 4 و 5.

الشجرة الفرعية للعقدة 0 تحتوي على كل العقد في الشجرة والتي عددها 6،

والشجرة الفرعية للعقدة 4 تحتوي فقط على العقدة 4.



يتم تخصيص قيم غير سالبة صحيحة لكل عقدة وهي تمثل الوزن لهذه العقدة.

نرمز لوزن العقدة i ($0 \leq i < N$) بـ $W[i]$.

يجب عليك كتابة برنامج للإجابة على Q من الطلبات، كل طلب يحدد بزوج من الأرقام الصحيحة الموجبة (L, R) .

الإجابة على الطلبات تحسب وفق التالي:

نعرف عدد صحيح نسميه **معامل** لكل عقدة من الشجرة.

هذا العدد يوصف بواسطة سلسلة $C[0], \dots, C[N-1]$ ، حيث $C[i]$ ($0 \leq i < N$) هو المعامل الخاص بالعقدة i .

نسمي هذه السلسلة بـ **سلسلة المعامل** نلاحظ أن العناصر داخل هذه السلسلة من الممكن أن تكون سالبة ، 0، أو موجبة.

لكل طلب (L, R) ، تكون سلسلة المعامل صالحة إذا كانت كل عقدة i ($0 \leq i < N$)، تحقق الشرط التالي:

مجموع المعاملات للعقد الموجودة في الشجرة الفرعية للعقدة i ، ليس أقل من L وليس أكبر من R .

لكل سلسلة معامل معطاة $C[0], \dots, C[N-1]$ ، **التكلفة** للعقدة i هي $|C[i]| \cdot W[i]$ ، حيث $|C[i]|$ يرمز للقيمة المطلقة لـ $C[i]$.

أخيراً، **التكلفة الكلية** هي مجموع التكاليف لكل العقد.

يجب عليك حساب **أقل تكلفة** يمكن الوصول إليها عند أي سلسلة معامل صالحة بالنسبة لكل طلب.

من الواضح وجود سلسلة معامل صحيحة واحدة على الأقل بالنسبة لأي طلب.

تفاصيل البرمجة

يجب عليك برمجة الدالتين التاليين:

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

- P, W : مصفوفة من الأعداد الصحيحة بطول N تحدد الآباء والأوزان.
- يتم استدعاء هذه الدالة مرة واحدة فقط في بداية عملية التواصل بين نظام التصحيح والبرنامج الخاص بك لكل حالة اختبار.

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

- L, R : أعداد صحيحة تمثل الطلب.
- هذه الدالة تستدعي Q بعد استدعاء الدالة `init` لكل حالة اختبار.
- يجب أن تعيد هذه الدالة جواب الطلب.

الحدود

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ من أجل كل i حيث $1 \leq i < N$
- $0 \leq W[i] \leq 1\,000\,000$ من أجل كل i حيث $0 \leq i < N$
- $1 \leq L \leq R \leq 1\,000\,000$ في كل طلب

المسائل الجزئية

المسألة الجزئية	العلامة	قيود إضافية
1	10	$Q \leq 10; W[P[i]] \leq W[i]$ أن i حيث $1 \leq i < N$
2	13	$Q \leq 10; N \leq 2\,000$
3	18	$Q \leq 10; N \leq 60\,000$
4	7	$W[i] = 1$ أن i حيث $0 \leq i < N$
5	11	$W[i] \leq 1$ أن i حيث $0 \leq i < N$
6	22	$L = 1$
7	19	بدون قيود إضافية.

الأمثلة

لتكن الاستدعاءات التالية:

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

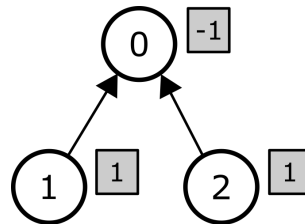
تتكون الشجرة من 3 عقد، الجذر وأطفالها الـ 2. وزن كل العقد 1.

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

في هذا الطلب $L = R = 1$ ، هذا يعني أن مجموع المعاملات في كل شجرة الفرعية يجب أن تكون 1.

لتكن سلسلة المعامل $[-1, 1, 1]$.

الشجرة والمعاملات المقابلة لها (في المستطيلات المظللة) موضحة أدناه.



من أجل كل عقدة i ($0 \leq i < 3$)، مجموع المعاملات لكل العقد في الشجرة الفرعية للعقدة i تساوي 1.

إذاً، سلسلة المعامل هذه صالحة.

تحسب التكلفة بالشكل التالي:

العقدة	الوزن	المعامل	التكلفة
0	1	-1	$ -1 \cdot 1 = 1$
1	1	1	$ 1 \cdot 1 = 1$
2	1	1	$ 1 \cdot 1 = 1$

لذلك التكلفة هي 3. هذه سلسلة المعامل الوحيدة الصالحة، حيث يجب على الدالة أن ترجع قيمة 3.

query(1, 2)

أصغر تكلفة إجمالية لهذا الطلب هي 2، وهي تحدث عندما تكون سلسلة المعامل هي $[0, 1, 1]$.

Sample Grader

:Input format

```
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]
```

where $L[j]$ and $R[j]$ (for $0 \leq j < Q$) are the input arguments in the j -th call to query. Note that the second line of the input contains **only** $N - 1$ **integers**, as the sample grader does not read the $P[0]$ value of

:Output format

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

.where $A[j]$ (for $0 \leq j < Q$) is the value returned by the j -th call to query