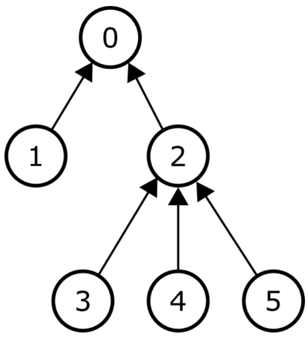
עץ

קיים **עץ** בעל N **צמתים**, שממוספרים מ-0 עד $N-1$. לצומת מספר 0 קוראים **השורש**. לכל צומת, חוץ מהשורש, יש **אב** יחיד. עבור כל i שמקיים $1 \leq i < N$, האב של הצומת ה-i הוא הצומת $P[i]$, כאשר $P[i] < i$. בנוסף אנחנו מניחים שמתקיים $P[0] = -1$.

עבור כל צומת i ($0 \leq i < N$), **תת העץ** של i הוא הקבוצה של הצמתים הבאים:

- i, וגם
- כל צומת שהאב שלו הוא i, וגם
- כל צומת שהאב של האב שלו הוא i, וגם
- כל צומת שהאב של האב של האב שלו הוא i, וגם
- וכו'.

התמונה למטה מציגה דוגמה לעץ בו $N = 6$ צמתים. כל חץ מחבר צומת אל האב שלו, חוץ מהשורש, שאין לו אב. תת העץ של הצומת 2 מכיל את הצמתים $2, 3, 4$ ו-5. תת העץ של הצומת 0 מכיל את כל 6 הצמתים של העץ ותת העץ של הצומת 4 מכיל רק את הצומת 4.



לכל צומת יש **משקל** שהוא מספר שלם אי-שלילי. המשקל של הצומת ה-i ($0 \leq i < N$) מיוצג על ידי $W[i]$.

משימתכם היא לכתוב תוכנית שתענה על Q שאילתות, כל אחת תיוצג על ידי זוג מספרים שלמים חיוביים (L, R). התשובה לשאילתה תחושב באופן הבא.

עבור כל צומת בעץ, אפשר לבחור מספר שלם, הנקרא **מקדם**. בחירה כזאת מתוארת על ידי הרצף $C[0], \ldots, C[N-1]$, כאשר $C[i]$ ($0 \leq i < N$) הוא המקדם המתאים לצומת ה-i. נקרא לרצף זה **רצף מקדמים**. שימו לב שהאיברים ברצף המקדמים יכולים להיות שליליים, 0, או חיוביים.

עבור השאילתה (L, R), רצף מקדמים נקרא **תקין** אם, עבור כל צומת i ($0 \leq i < N$), מתקיים התנאי הבא: סכום המקדמים של הצמתים בתת העץ של הצומת ה-i הוא לא קטן מ-L ולא גדול מ-R.

בהינתן רצף מקדמים $C[0], \ldots, C[N-1]$, ה**עלות** של הצומת ה-i היא $|C[i]| \cdot W[i]$, כאשר $|C[i]|$ מייצג את הערך המוחלט של $C[i]$. לבסוף, ה**עלות הכוללת** היא סכום העלויות של כל הצמתים. המשימה שלכם היא לחשב,

עבור כל שאילתה, את **העלות הכוללת המינימלית** שיכולה להתקבל על ידי רצף מקדמים תקין כלשהו.

ניתן להראות שעבור כל שאילתה, לפחות רצף מקדמים תקין אחד קיים.

פרטי מימוש

עליכם לממש את הפונקציות הבאות:

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

- P, W: מערכים באורך N של מספרים שלמים המתארים את האבות והמשקלים.
- פונקציה זו נקראת בדיוק פעם אחת בתחילת האינטרקציה בין הגריידר והתוכנית שלכם בכל טסט.

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

- L, R: מספרים שלמים המתארים שאילתה.
- פונקציה זו נקראת Q פעמים לאחר הריצה של `init` בכל טסט.
- פונקציה זו צריכה להחזיר את התשובה לשאילתה שניתנה.

מגבלות

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ עבור כל i שמקיים $1 \leq i < N$
- $0 \leq W[i] \leq 1\,000\,000$ עבור כל i שמקיים $0 \leq i < N$
- $1 \leq L \leq R \leq 1\,000\,000$ בכל שאילתה

תתי משימות

תת משימה	ניקוד	מגבלות נוספות
1	10	$Q \leq 10$; $W[P[i]] \leq W[i]$ עבור כל i שמקיים $1 \leq i < N$
2	13	$Q \leq 10$; $N \leq 2\,000$
3	18	$Q \leq 10$; $N \leq 60\,000$
4	7	$W[i] = 1$ עבור כל i שמקיים $0 \leq i < N$
5	11	$W[i] \leq 1$ עבור כל i שמקיים $0 \leq i < N$
6	22	$L = 1$
7	19	ללא מגבלות נוספות.

דוגמאות

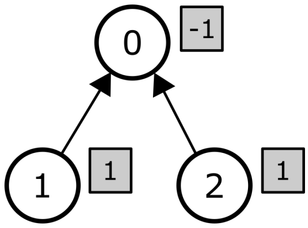
בהינתן הקריאות הבאות:

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

העץ מכיל 3 צמתים, השורש ו-2 בניו. לכל הצמתים יש משקל 1.

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

בשאילתה הזאת $L = R = 1$, כלומר סכום המקדמים בכל תת עץ חייב להיות שווה 1. בהינתן רצף המקדמים $[-1, 1, 1]$, העץ והמקדמים המתאימים (במלבנים המוצללים) מאוירים למטה.



עבור כל צומת i ($0 \leq i < 3$), סכום המקדמים של כל הצמתים בתת העץ של i שווה ל-1. לכן, רצף המקדמים הזה הוא תקין. העלות הכוללת מחושבת באופן הבא:

צומת	משקל	מקדם	עלות
0	1	-1	$\mid -1 \mid \cdot 1 = 1$
1	1	1	$\mid 1 \mid \cdot 1 = 1$
2	1	1	$\mid 1 \mid \cdot 1 = 1$

לפיכך העלות הכוללת היא 3. זה רצף המקדמים התקין היחיד, לכן הקריאה הזאת אמורה להחזיר 3.

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

העלות הכוללת המינימלית עבור השאילתה הזאת היא 2, וניתן להשיג זאת כאשר רצף המקדמים הוא $[0, 1, 1]$.

גריידר לדוגמה

פורמט הקלט:

```
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]
```

כאשר $L[j]$ ו- $R[j]$ (עבור $0 \leq j < Q$) הם משתני הקלט בקריאה ה-j ל-`query`. שימו לב שהשורה השנייה בקלט מכילה **רק** $N-1$ **מספרים שלמים**, מפני שהגריידר לדוגמה לא קורא את הערך של $P[0]$.

פורמט הפלט:

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

כאשר $A[j]$ (עבור $0 \leq j < Q$) הערך המוחזר על ידי הקריאה ה-j ל-`query`.