

Diametro Dinamico (diameter)

Giorno	1
Lingua	Italiano
Limite di tempo:	5 secondi
Limite di memoria:	1024 megabyte

Ti viene fornito un albero pesato di n vertici con archi non direzionati e una lista di q update. Ogni update cambia il peso di un arco. L'obiettivo è stampare in output il diametro dell'albero dopo ogni update. (La distanza tra due vertici è la somma dei pesi dell'unico percorso che li collega. Il diametro è la più grande di queste distanze.)

File di input

La prima linea contiene tre interi separati da spazi n , q e w ($2 \leq n \leq 100,000$, $1 \leq q \leq 100,000$, $1 \leq w \leq 20,000,000,000,000$) – il numero di vertici dell'albero, il numero di update e il limite sui pesi degli archi. I vertici sono numerati da 1 a n .

Le successive $n - 1$ righe descrivono l'albero iniziale. L' i -esima di queste righe contiene tre interi separati da spazi a_i , b_i , c_i ($1 \leq a_i, b_i \leq n$, $0 \leq c_i < w$) che specificano che inizialmente esiste un arco tra i vertici a_i e b_i con peso c_i . È garantito che queste $n - 1$ righe descrivono un albero.

Infine seguono q righe che descrivono le query. La j -esima di queste righe contiene due interi separati da spazi d_j , e_j ($0 \leq d_j < n - 1$, $0 \leq e_j < w$). Questi due interi sono trasformati secondo il seguente schema:

- $d'_j = (d_j + last) \bmod (n - 1)$
- $e'_j = (e_j + last) \bmod w$

dove $last$ è il risultato dell'ultima query (inizialmente $last = 0$). La tupla (d'_j, e'_j) rappresenta una query che prende il $d'_j + 1$ -esimo arco secondo l'ordine di input e cambia il suo peso in e'_j .

File di output

Dovrai stampare q righe. Per ogni i , la riga i dovrà contenere il diametro dell'albero dopo l' i -esimo update.

Assegnazione del punteggio

Subtask 1 (11 punti): $n, q \leq 100$ e $w \leq 10,000$

Subtask 2 (13 punti): $n, q \leq 5,000$ e $w \leq 10,000$

Subtask 3 (7 punti): $w \leq 10,000$ e gli archi dell'albero sono esattamente tutti archi validi nella forma $\{1, i\}$ (In pratica l'albero è una stella centrata al vertice 1.)

Subtask 4 (18 punti): $w \leq 10,000$, e gli archi dell'albero sono esattamente tutti archi validi nella forma $\{i, 2i\}$ e $\{i, 2i + 1\}$ (In pratica, se considerassimo il vertice 1 come la radice dell'albero, sarebbe un albero binario bilanciato.)

Subtask 5 (24 punti): è garantito che dopo ogni update il percorso semplice più lungo passa per il vertice 1.

Subtask 6 (27 punti): nessuna limitazione.



Quindi il peso dell'arco $\{1, 2\}$ è cambiato a 1050. Questo porta la coppia di vertici $\{1, 4\}$ ad essere la coppia a maggiore distanza, con valore 2080.

La terza query è decodificata come:

$$\begin{aligned}d'_3 &= (1 + 2080) \bmod 3 = 2 \\e'_3 &= (890 + 2080) \bmod 2000 = 970\end{aligned}$$

Dal momento che la distanza dei vertici $\{2, 4\}$ diminuisce a 970, la coppia pi distante improvvisamente $\{1, 3\}$ con un valore di 2050.