Dynamischer Durchmesser (diameter)

Tag	1
Sprache	Deutsch
Zeitlimit:	5 Sekunden
Speicherlimit:	1024 Megabytes

Du erhältst einen gewichteten, ungerichteten Baum mit n Knoten und eine Liste von q Updates. Jedes Update ändert das Gewicht einer Kante. Die Aufgabe ist es, den Durchmesser des Baumes nach jedem Update auszugeben.

(Die Distanz zweier Knoten ist die Summe der Gewichte auf dem eindeutigen simplen Pfad, welcher diese verbindet. Der Durchmesser ist die grösste all dieser Distanzen.)

Eingabe

Die erste Zeile enthält drei leerzeichengetrennte Ganzzahlen n, q und w ($2 \leq n \leq 100\,000, 1 \leq q \leq 100\,000$, $1 \leq w \leq 20\,000\,000\,000\,000$) – die Anzahl Knoten im Baum, die Anzahl Updates und das Limit für die Kantengewichte. Die Knoten sind von 1 bis n durchnummeriert.

Die nächsten $n-1$ Zeilen beschreiben den Anfangszustand des Baums. Die i-te dieser Zeilen enthält drei leerzeichengetrennte Ganzzahlen a_i, b_i, c_i ($1 \leq a_i, b_i \leq n$, $0 \leq c_i < w$) und bedeutet, dass es eine Kante zwischen den Knoten a_i und b_i mit anfänglichem Gewicht c_i gibt. Es ist garantiert, dass diese $n-1$ Zeilen einen Baum beschreiben.

Es folgen q Zeilen, welche die Abfragen beschreiben. Die j-te dieser Zeilen enthält zwei leerzeichengetrennte Ganzzahlen d_j, e_j ($0 \leq d_j < n-1, 0 \leq e_j < w$). Diese zwei Ganzzahlen werden dann wie folgt transformiert:

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

wobei $last$ für das Ergebnis der letzten Abfrage steht (zu Beginn ist $last = 0$). Das Tupel (d'_j, e'_j) repräsentiert eine Abfrage, welche das Gewicht der $d'_j + 1$-ten Kante auf e'_j setzt.

Ausgabe

Gib q Zeilen aus. Die i-te Zeile soll dabei jeweils den Durchmesser des Baumes nach dem i-ten Update enthalten.

Bewertung

Teilaufgabe 1 (11 Punkte): $n, q \leq 100$ und $w \leq 10\,000$.

Teilaufgabe 2 (13 Punkte): $n, q \leq 5\,000$ und $w \leq 10\,000$.

Teilaufgabe 3 (7 Punkte): $w \leq 10\,000$ und die Kanten des Baumes sind genau alle gültigen Kanten der Form $\{1, i\}$ (d.h. der Baum ist ein Stern mit Zentrum im Knoten 1.)

Teilaufgabe 4 (18 Punkte): $w \leq 10\,000$ und die Kanten des Baumes sind genau alle gültigen Kanten der Form $\{i, 2i\}$ und $\{i, 2i+1\}$ (d.h. würde man den Baum bei Knoten 1 wurzeln, entstünde ein balancierter Binärbaum.)

Teilaufgabe 5 (24 Punkte): Es ist garantiert, dass nach jedem Update ein längster einfacher Pfad durch Knoten 1 geht.

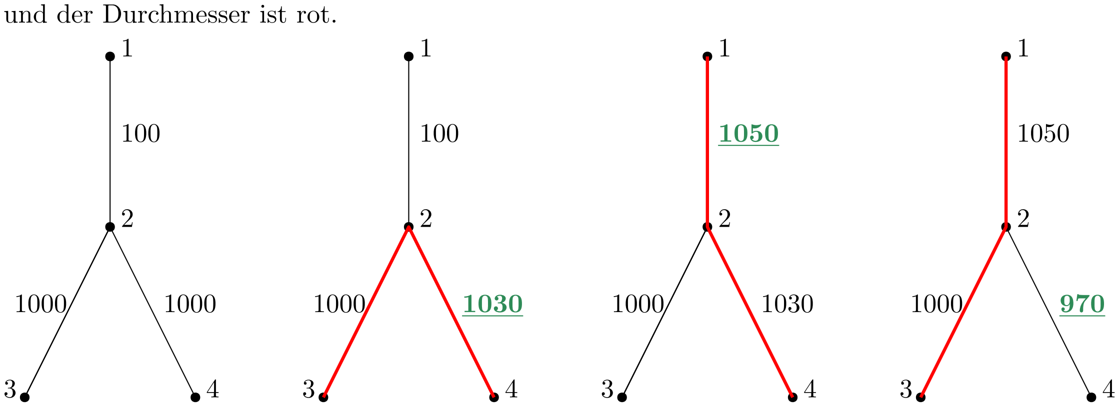
Teilaufgabe 6 (27 Punkte): Keine weiteren Einschränkungen.

Beispiele

standard input	standard output
4 3 2000 1 2 100 2 3 1000 2 4 1000 2 1030 1 1020 1 890	2030 2080 2050
10 10 10000 1 9 1241 5 6 1630 10 5 1630 2 6 853 10 1 511 5 3 760 8 3 1076 4 10 1483 7 10 40 8 2051 5 6294 5 4168 7 1861 0 5244 6 5156 3 3001 8 5267 5 3102 8 3623	6164 7812 8385 6737 6738 7205 6641 7062 6581 5155

Bemerkung

Das erste Beispiel ist unten abgebildet. Das Bild ganz links zeigt den initialen Zustand des Graphs. Jedes der folgenden Bildern zeigt den Zustand nach einem Update. Das gewicht der aktualisierten Kante ist grün gezeichnet und der Durchmesser ist rot.



Die erste Abfrage ändert das Gewicht der 3-ten Kante, also $\{2, 4\}$, auf 1030. Die grösste Distanz zwischen zwei Knoten ist 2030 – die Distanz zwischen 3 und 4.

Da die Antwort 2030 ist, ergibt sich die zweite Abfrage als

$$d'_2 = (1 + 2030) \bmod 3 = 0$$

$$e'_2 = (1020 + 2030) \bmod 2000 = 1050$$

Daher wird das Gewicht der Kante $\{1, 2\}$ auf 1050 geändert. Dies bewirkt, dass das Paar $\{1, 4\}$ nun die grösste Distanz hat, nämlich 2080.

Die dritte Abfrage wird folgendermassen entziffert:

$$d'_3 = (1 + 2080) \bmod 3 = 2$$

$$e'_3 = (890 + 2080) \bmod 2000 = 970$$

Da das Gewicht der Kante $\{2, 4\}$ sich auf 970 verringert ist das am weitesten entfernte Paar plötzlich $\{1, 3\}$ mit 2050.