Diametrul Dinamic (diameter)

Day	1
Language	Romanian
Time limit:	5 seconds
Memory limit:	1024 megabytes

Vi se da un arbore cu costuri pe muchii de n noduri si o lista de q update-uri. Fiecare update schimba costul unei muchii. Cerinta este sa se afiseze diametrul arborelui dupa fiecare update.

(Distanta dintre 2 noduri este suma costurilor pe drumul simplu si unic care le conecteaza. Diametrul este maximul dintre aceste distante.)

Input

Pe prima linie se afla trei numere intregi separate prin spatiu n, q si w ($2 \leq n \leq 100,000, 1 \leq q \leq 100,000$, $1 \leq w \leq 20,000,000,000,000$) – numarul de noduri din arbore, numarul de update-uri si costul maxim posibil al unei muchii. Nodurile sunt numerotate de la 1 la n.

Urmatoarele $n-1$ linii descriu starea initiala a arborelui. A i-a linie dintre acestea contine 3 numere intregi separate prin spatii: a_i, b_i, c_i ($1 \leq a_i, b_i \leq n$, $0 \leq c_i < w$) semnificand ca initial exista o muchie intre nodurile a_i si b_i de cost c_i. Este garantat ca aceste $n-1$ linii definesc un arbore.

Ultimele q linii descriu query-urile. A j-a dintre acestea contine doua numere intregi separate prin spatiu: d_j, e_j ($0 \leq d_j < n-1, 0 \leq e_j < w$). Aceste doua numere se transforma conform urmatoarelor formule:

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

unde $last$ este rezultatul ultimului query (initial $last = 0$). Perechea (d'_j, e'_j) reprezinta un query care schimba costul muchiei cu indicele $d'_j + 1$, setandu-i costul la e'_j.

Output

Afisati q linii. Pentru fiecare i, linia i trebuie sa contina diametrul arborelui dupa update-ul cu indicele i.

Scoring

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

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

Subtask 3 (7 puncte): $w \leq 10,000$ si toate muchiile arborelui sunt de forma $\{1, i\}$ (Practic, arborele este o stea centrata in nodul 1.)

Subtask 4 (18 puncte): $w \leq 10,000$, si toate muchiile arborelui sunt de forma $\{i, 2i\}$ si $\{i, 2i+1\}$ (Prin urmare, daca am fixa radacina in nodul 1, acesta ar fi un arbore binar balansat.)

Subtask 5 (24 puncte): este garantat ca toate update-urile care modifica cel mai lung drum simplu trec prin nodul 1

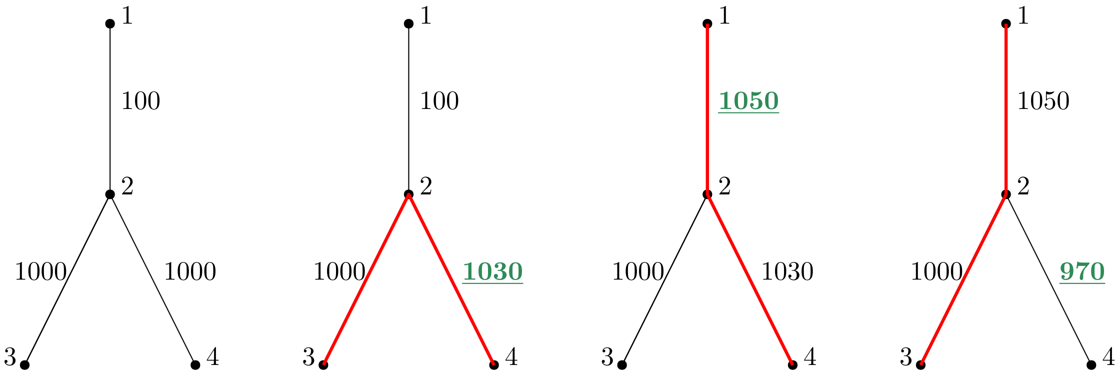
Subtask 6 (27 puncte): nicio restrictie suplimentara

Examples

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

Note

Primul exemplu este reprezentat in figura de mai jos. Cea mai din stanga imagine prezinta starea initiala a grafului. Fiecare dintre imaginile urmatoare reprezinta starea de dupa update. Costul muchiei careia i s-a facut update-ul este colorat in verde, iar diametrul in rosu.



Primul query modifica costul celei de a 3-a muchii, i.e. $\{2, 4\}$ in 1030. Distanta maxima dintre oricare 2 perechi de noduri este 2030 – distanta dintre nodurile 3 si 4.

Avand acum raspunsul 2030, al doilea query este

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

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

Prin urmare, costul muchiei $\{1,2\}$ este modificat in 1050. Aceasta cauzeaza ca perechea $\{1,4\}$ sa devina perechea ce are distanta maxima, anume 2080.

Al treilea query se decodifica astfel:

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

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

Astfel costul muchiei $\{2,4\}$ este scazut la 970, cele mai departate noduri ajung $\{1,3\}$ cu cost 2050.