

Problema Vasile

Fișier de intrare `stdin`
Fișier de ieșire `stdout`

Banii mulți se fac ușor, banii puțini se fac greu.

— 'Claus

Vasile este prietenul [Linei](#). Recent, Vasile și-a făcut un cont de Pinance pentru a petrece mai mult timp investind în criptomonede decât la biserică. Surpriza lui a venit într-o zi când, fără vreun motiv aparent, a observat că investițiile lui au dat roade. Surprins de acest eveniment, a decis să analizeze evenimentele ce au avut loc în Plockchain care l-au adus la aceste culmi. Vasile ne-a explicat că ne putem imagina că Plockchain-ul funcționează în felul următor:

La început, în sistem există N cereri de tranzacții, indexate de la 1 la N , a i -a din ele având un comision la efectuare de W_i și o unică altă cerere de care este direct dependentă P_i . Se spune că cererea cu indexul 1 era primordială sistemului și nu are nicio dependență. Vasile a observat mișcările din Plockchain și le-a clasificat în trei tipuri:

1. Apare o nouă cerere cu comision inițial C , a cărei dependență directă este T . Indexul acestei cereri va fi $N + t + 1$, unde t este numărul de cereri adăugate anterior prin acest procedeu. Astfel, prima cerere adăugată are indexul $N + 1$.
2. Este modificat comisionul unei cereri cu index a din W_a în C .
3. Este ștearsă complet din registru cererea cu index a . Se garantează că nu există o altă cerere j astfel încât $P_j = a$ (nu depindea nicio cerere de a).

[Lina](#), uimită de explicațiile lui Vasile, a început să se întrebe adesea de-a lungul acestor modificări următoarea întrebare:

Considerând toate cererile care există acum în sistem, cu ce comision ar fi răsplătit un hacker dacă ar efectua toate cererile care se află la cel mult H dependențe în jos față de o cerere X ?

Spunem că o cerere A se află la cel mult x dependențe în jos față de B dacă și numai dacă $x \geq 0$ și $A = B$, sau $x \geq 1$ și P_A se află la cel mult $x - 1$ dependențe în jos față de B . Considerăm de asemenea că, în urma efectuării unor cereri, hackerul primește suma comisioanelor aferente cererilor. Desigur, întrebările [Linei](#) nu au niciun efect asupra sistemului, acesta rămânând la fel în urma acestei operații. Din vina sistemului robust al Plockchain-ului, H este același în orice întrebare de-a [Linei](#).

Cerință

Vasile știe că ziua [Linei](#) va fi curând, așa că s-a gândit să-i ofere cel mai bun cadou din lume: O listă de numere ce reprezintă răspunsurile la întrebările ei. Cu toate acestea, el este mult prea ocupat să o facă singur, așa că te-a delegat pe tine să le afli.

Date de intrare

Pe primul rând se află numerele N , Q și H , reprezentând numărul de cereri din sistemul original, numărul de modificări explicate de Vasile sau întrebări de-a [Linei](#), și constanta H din întrebările [Linei](#).

Urmează pe al doilea rând N numere, al i -lea reprezentând W_i , comisionul inițial al cererii cu indexul i .

Urmează pe al treilea rând $N - 1$ numere, al i -lea reprezentând P_{i+1} , cererea de care este direct dependentă cererea cu indexul $i + 1$.

Urmează apoi Q rânduri, primul număr de pe fiecare fiind t , reprezentând tipul operației. Apoi:

1. Dacă $t = 1$, urmează C și T , reprezentând parametrii cererii nou apărute.

2. Dacă $t = 2$, urmează a și M , reprezentând indexul cererii a cărei comision se modifică respectiv valoarea în care se modifică.
3. Dacă $t = 3$, urmează a , reprezentând indexul cererii care este acum ștersă din sistem.
4. Dacă $t = 4$, urmează X , reprezentând indexul cererii pentru care pune [Lina](#) întrebarea.

Date de ieșire

În urma fiecărei întrebări (operație cu $t = 4$), se va afișa un singur număr, anume răspunsul la întrebarea [Linei](#) pentru sistemul curent.

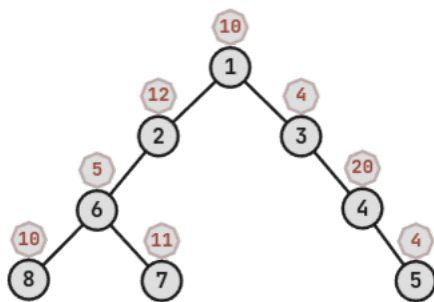
Restricții

- $2 \leq N \leq 200\,000$
- $1 \leq Q \leq 200\,000$
- $1 \leq H \leq N$
- $0 \leq C, M \leq 1\,000\,000\,000$ pentru toate operațiile.
- $T < R$ pentru toate operațiile de tipul $t = 1$, unde cu R am notat indexul cererii nou adăugate în urma operației.
- $1 \leq P_i < i$ și $0 \leq W_i \leq 1\,000\,000\,000$, pentru orice cerere inițială i .
- Se garantează că pentru orice operație cererile implicate în aceasta există la momentul efectuării operației.
- Se garantează că nicio operație de tipul $t = 3$ va fi efectuată asupra cererii cu numărul 1.
- Se garantează că nicio operație de tipul $t = 3$ va fi efectuată asupra unei cereri i pentru care există la momentul efectuării un j astfel încât $P_j = i$.

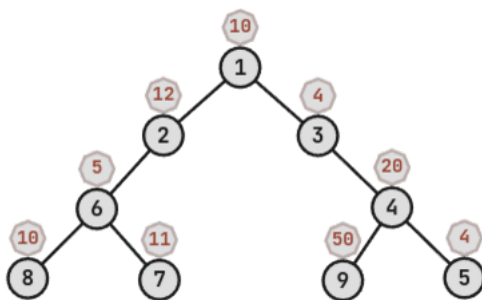
#	Punctaj	Restricții
1	13	$N, Q \leq 5\,000$
2	11	$P_i = i - 1$, pentru $2 \leq i \leq N$, iar $t = 4$ pentru orice operație
3	16	$t = 4$ pentru orice operație
4	10	$H = 1$
5	14	$P_i = i - 1$, pentru $2 \leq i \leq N$, iar $t = 2$ sau $t = 4$ pentru orice operație
6	10	$P_i = \lfloor \frac{i}{2} \rfloor$, pentru $2 \leq i \leq N$, iar $t = 2$ sau $t = 4$ pentru orice operație
7	26	Nu există restricții suplimentare

Exemple

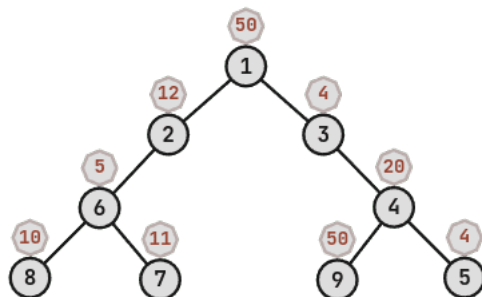
stdin	stdout	Explicații
8 7 2	78	Pentru prima interogare, cererile care sunt la 2 dependențe în jos față de cererea 3 sunt 3, 4, 9, 5 cu costurile 4, 20, 50, 4. Astfel, trebuie să afișăm $4 + 20 + 50 + 4 = 78$.
10 12 4 20 4 5 11	91	
10	61	
1 1 3 4 2 6 6		
1 50 4		
4 3		
2 1 50		
4 1		
2 6 35		
1 5 7		
4 6		



Structura inițială



Structura după prima operație



Structura după primele 3 operații