

Problema Arbore

Fișier de intrare `arbore.in`
Fișier de ieșire `arbore.out`

Se dă un arbore cu N noduri, numerotate de la 1 la N . Fiecare muchie din arbore are un cost număr natural nenul asociat. O operație constă în a alege o muchie cu costul **strict** pozitiv și a-i scădea costul cu 1.

Pentru un nod v din arbore și un număr natural k , se notează cu $f(v, k)$ suma minimă a costurilor drumurilor de la nodul v la restul nodurilor din arbore care se poate obține aplicând în mod optim **cel mult** k operații de tipul descris anterior.

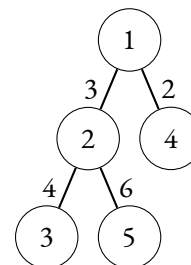


Figura 1

Cerință

Se dau Q interogări de forma (v, k_1, k_2) , unde v este un nod din arbore, iar k_1 și k_2 sunt numere naturale astfel încât $k_1 \leq k_2$. Pentru fiecare interogare să se calculeze $\sum_{k=k_1}^{k_2} f(v, k)$, adică suma valorilor $f(v, k)$ pentru $k_1 \leq k \leq k_2$, modulo $10^9 + 7$.

Date de intrare

Prima linie a fișierului de intrare `arbore.in` conține numărul N , reprezentând numărul de noduri din arbore.

Pe următoarele $N - 1$ linii se află câte 3 numere $u \ v \ w$, reprezentând o muchie din arbore între nodurile u și v , care are costul w .

Următoarea linie conține un singur număr Q , reprezentând numărul de interogări, iar pe următoarele Q linii se află câte trei numere $v \ k_1 \ k_2$, cu semnificația din enunț.

Date de ieșire

Fișierul de ieșire `arbore.out` trebuie să conțină Q linii, fiecare conținând câte un număr natural care reprezintă răspunsul pentru fiecare interogare dată, în ordinea în care acestea sunt date în fișierul de intrare.

Restricții

- $2 \leq N \leq 200\,000$
- $1 \leq Q \leq 400\,000$
- $1 \leq w \leq 10^8$
- $0 \leq k_1 \leq k_2 \leq 10^{14}$

#	Punctaj	Restricții
1	12	$N \leq 2\,000, Q \leq 2\,000, k_1 = k_2 = 0$
2	7	$N \leq 200\,000, Q \leq 400\,000, k_1 = k_2 = 0$
3	13	$N \leq 2\,000, Q \leq 2\,000, k_1 = k_2$
4	16	$N \leq 5\,000, Q \leq 200\,000, k_1 = k_2$
5	19	$N \leq 100\,000, Q \leq 100\,000, k_1 = k_2$
6	14	$N \leq 200\,000, Q \leq 400\,000, k_1 = k_2$
7	11	$N \leq 100\,000, Q \leq 100\,000$
8	8	Fără restricții suplimentare.

Exemplu

arbore.in	arbore.out
5	21
1 2 3	16
2 3 4	0
2 5 6	27
1 4 2	
4	
1 4 5	
2 1 1	
5 20 22	
4 0 0	

Explicația exemplului

Arborele are $N = 5$ noduri și este ilustrat în Figura 1 împreună cu costurile asociate muchiilor sale. Se dau $Q = 4$ interogări, după cum urmează.

Prima interogare: Pentru $k = 4$ se poate aplica operația descrisă în enunț muchiei 1–2 de 3 ori, respectiv muchiei 1–4 o dată, deci $f(1, 4) = 11$. Pentru $k = 5$ se poate aplica operația descrisă în enunț muchiei 1–2 de 3 ori, muchiei 2–3 o dată, respectiv muchiei 2–5 o dată, deci $f(1, 5) = 10$. Astfel, răspunsul este $f(1, 4) + f(1, 5) = 11 + 10 = 21$.

A doua interogare: Se poate scădea costul muchiei 1–2 o dată, deci răspunsul este $f(1, 1) = 16$.

A treia interogare: Se observă că pentru oricare $20 \leq k \leq 22$, costurile tuturor muchiilor din arbore pot fi scăzute până ajung la valoarea 0 (mai mult de atât nu este permis din definiția operației date), deci rezultatul este 0.

A patra interogare: $k_1 = k_2 = 0$, deci nu avem la dispoziție nicio operație care să scadă costul muchiilor. Astfel, răspunsul este dat de suma costurilor drumurilor de la nodul 4 la restul nodurilor din arbore, care este $2 + 5 + 9 + 11 = 27$.

Problema Detonator

Fișier de intrare `detonator.in`
Fișier de ieșire `detonator.out`

Comisarul Roman se află în fața unui dispozitiv exploziv constând dintr-o piramidă cu N nivele numerotate de la 1 la N . Fiecare nivel i conține i bombe numerotate de la 1 la i . Notăm bomba j de pe nivelul i cu $B_{i,j}$. Pentru fiecare bombă $B_{i,j}$ se cunoaște timpul în secunde $T_{i,j}$ de la momentul inițial după care aceasta explodează. Roman trebuie să dezamorseze dispozitivul după următoarele reguli:

1. Dezamorsarea unei bombe durează o secundă;
2. Pentru a dezamorsa bomba $B_{i,j}$ trebuie întâi dezamorsate cele două bombe peste care aceasta este așezată, adică $B_{i+1,j}$ și $B_{i+1,j+1}$. Bombele de pe nivelul N nu au bombe dedesubt și deci nu se supun acestei reguli.



Figura 1: Exemplu pentru $N = 4$. Valorile $T_{i,j}$ sunt înscrise pe bombe.

Dispozitivul se consideră dezamorsat odată ce toate bombele au fost dezamorsate. Roman nu vrea să se grăbească, așa că ar vrea să știe care este numărul maxim de secunde X astfel încât, dacă ar începe operațiunea de dezamorsare cu o întârziere inițială de X secunde, dispozitivul ar putea fi încă dezamorsat cu succes. Cu alte cuvinte, se cere cel mai mare număr X astfel încât dezamorsarea ar fi posibilă dacă am înlocui numerele $T_{i,j}$ cu $T_{i,j} - X$ pentru $1 \leq j \leq i \leq N$. Este posibil și ca X să fie negativ dacă Roman a ajuns prea târziu la dispozitiv. De exemplu, dacă dezamorsarea nu este posibilă în condițiile date dar ar fi fost posibilă dacă s-ar fi ajuns la fața locului cu o secundă mai devreme, atunci $X = -1$. Dacă nici cu o secundă mai devreme nu s-ar fi putut, dar s-ar fi putut cu două secunde mai devreme, atunci $X = -2$, și așa mai departe.

Cerință

Pentru Q teste, date fiind N și valorile $T_{i,j}$ pentru $1 \leq j \leq i \leq N$, se cere numărul X .

Date de intrare

Prima linie a fișierului de intrare `detonator.in` conține Q , reprezentând numărul de teste. Urmăază descrierea celor Q teste, fiecare test fiind descris după cum urmează. Prima linie conține numărul întreg N . Următoarele N linii, numerotate în cadrul testului de la 1 la N , conțin numere întregi astfel încât al j -lea număr de pe linia i este $T_{i,j}$. Observați că linia i conține i numere.

Date de ieșire

Fișierul de ieșire `detonator.out` trebuie să conțină Q linii, reprezentând numărul X pentru fiecare test dat.

Restricții

- $1 \leq Q \leq 5$
- $1 \leq N \leq 1\,000$
- $1 \leq T_{i,j} \leq 10^9$ pentru $1 \leq j \leq i \leq N$.

#	Punctaj	Restricții
1	7	Valorile $T_{i,j}$ pentru $1 \leq j \leq i \leq N$ sunt egale (toate bombele sunt programate să explodeze în același timp).
2	13	$N \leq 5, X \leq 10$
3	9	$N \leq 5$
4	14	$N \leq 50, X \leq 10$ și $T_{i,j} \geq \max\{T_{i+1,j}, T_{i+1,j+1}\}$ pentru $1 \leq j \leq i < N$
5	13	$N \leq 50, X \leq 10$
6	2	$N \leq 50$
7	9	$N \leq 200, X \leq 10$
8	2	$N \leq 200$
9	19	$N \leq 500$
10	12	Fără restricții suplimentare.

Exemple

detonator.in	detonator.out	Explicații
4	7	Primul test are $N = 2$, iar cele $N \cdot (N + 1)/2 = 3$ bombe toate explodează după 10 secunde. Astfel, Roman poate aștepta maxim 7 secunde după care să înceapă procesul, dezamorsând cele trei bombe după 8, 9 și respectiv 10 secunde.
2	0	
10	-2	
10 10	0	
4		
10		
7 9		
4 6 8		
1 3 2 5		
3		
9		Al doilea test este ilustrat în Figura 1. În acest caz, Roman trebuie să înceapă dezamorsarea imediat, deoarece bomba $B_{4,1}$ ar exploda după o secundă. Singura ordine de dezamorsare validă este dată de valorile $B_{i,j}$ în ordinea $1, 2, \dots, 10$.
9 9		
1 1 1		
3		
6		
5 3		
4 4 4		
		Al treilea test are trei bombe ce explodează imediat, așa că Roman ar fi trebuit să ajungă cu 2 secunde mai devreme la dispozitiv.

Problema Zid

Fișier de intrare `zid.in`
Fișier de ieșire `zid.out`

Fascinată de cultura chineză și Marele Zid Chinezesc, Andra s-a hotărât să își construiască propriul zid în miniatură, de înălțime N și lățime M , din piese roșii și galbene pe care le deține.



Ea are la dispoziție un număr nelimitat de piese cu lățimea de o unitate și toate înălțimile posibile. Piese roșii (hóng) au înălțimea impară ($1, 3, 5, \dots$), pe când piesele galbene (huáng) au înălțimea pară ($2, 4, 6, \dots$). **Piese nu pot fi rotite și pot fi plasate doar pe verticală.**

Deoarece culoarea galbenă este considerată cea mai prestigioasă în cultura chineză, Andra vrea ca suma lungimilor tuturor pieselor galbene ce alcătuiesc zidul să fie egală cu un număr K , special ales astfel încât să aducă noroc. Mai mult de atât, ea se întreabă **în câte moduri** poate construi zidul astfel încât această condiție să fie respectată.

Întrucât această cerință “pare chineză” pentru ea (chiar dacă este vorbitoare de limba chineză!), a decis să vă ceară ajutorul, auzind că vă simțiți norocoși în anul dragonului.

Cerință

Date fiind N , M și K , determinați numărul de moduri de a construi zidul în condițiile date.

Date de intrare

Singura linie a fișierului de intrare `zid.in` conține valorile N , M și K , cu semnificația din enunț.

Date de ieșire

Singura linie a fișierului de ieșire `zid.out` va conține numărul de moduri de a construi zidul modulo $10^9 + 7$.

Restricții

- $1 \leq N, M \leq 5\,000$
- $0 \leq K \leq N$
- Două modalități de a construi zidul se consideră identice dacă sunt formate din aceleași tipuri de piese plasate în același mod. De exemplu, există două moduri de a construi un zid de 4×1 complet galben.

#	Punctaj	Restricții
1	10	$N \leq 6, M = 1$
2	16	$N \leq 500, M = 1$
3	5	$N \leq 2\,500, M = 1, K = N$
4	6	$N \leq 2\,500, M = 1, K = 0$
5	14	$N \leq 2\,500, M = 1$
6	4	$N \leq 500, M = 2$
7	2	$N \leq 500, M = 3$
8	5	$N \leq 2\,500, M = 4$
9	17	$N \leq 2\,500, M \leq 10$
10	14	$N \leq 2\,500, M \leq 2\,500$
11	7	$N \leq 5\,000, M \leq 5\,000$

Exemple

zid.in	zid.out	Explicații
5 1 2	6	Toate cele 6 configurații de ziduri cu înălțimea 5, lățimea 1 și $K = 2$ posibile sunt ilustrate în prima figură.
2 2 2	2	Toate cele 2 configurații sunt ilustrate în cea de a doua figură.

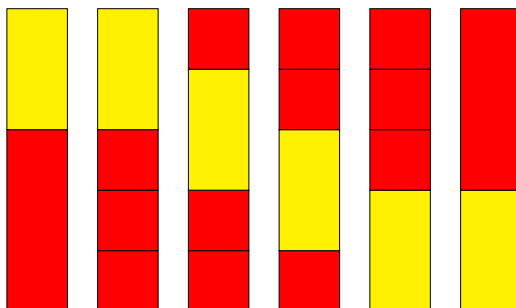


Figura 1

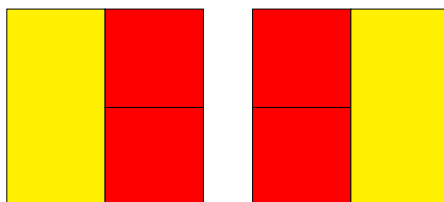


Figura 2