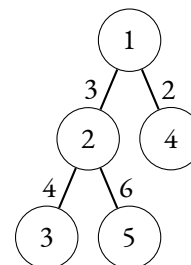


Feladat Arbore

Bemenet `arbore.in`
Kimenet `arbore.out`

Adott egy fa N csomóponttal, melyeket 1-től N -ig számozunk. A fa minden éléhez hozzá van rendelve egy költség, egy nullától különböző természetes szám. Egy művelet abban áll, hogy kiválasztunk egy élet, amelynek a költsége **szigorúan** pozitív, és csökkentjük az értékét 1-gyel.

A fa egy adott v csomópontjához, valamint egy természetes k szám esetén értelmezzük az $f(v, k)$ függvényt, mint a v csomópontból induló utak költségeinek minimális összegét, amelyet úgy érhetünk el, hogy az előzőleg leírt műveleteket optimálisan alkalmazzuk **legfeljebb** k alkalommal.



1. ábra

Követelmény

Adott Q lekérdezés (v, k_1, k_2) alakban, ahol v a fának egy csomópontja, illetve k_1 és k_2 természetes számok úgy, hogy $k_1 \leq k_2$. Minden lekérdezés esetén számítsátok ki a $\sum_{k=k_1}^{k_2} f(v, k)$ összeget, vagyis az $f(v, k)$ értékek összegét modulo $10^9 + 7$, minden k értékre, $k_1 \leq k \leq k_2$.

Bemeneti adatok

Az `arbore.in` bemeneti állomány első sora tartalmazza N értékét, a fa csomópontjainak számát.

A következő $N - 1$ sor mindegyikében található 3 szám: $u \ v \ w$, amely az u és v csomópontokat összekötő élet jelenti w költséggel.

A következő sor egyetlen Q számot tartalmaz, a lekérdezések számát, majd a következő Q sor mindegyike tartalmazni fog 3 számot, $v \ k_1 \ k_2$ értékeket, ahogyan meghatároztuk a kijelentésben.

Kimeneti adatok

Az `arbore.out` kimeneti állomány Q sort kell tartalmazzon, minden sorban egy természetes számot, a válaszokat a lekérdezésekre, a beolvasás sorrendjében.

Korlátok

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

#	Pontszám	Korlátok
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	Más megszorítások nincsenek.

Példa

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	

Példa magyarázat

A fának $N = 5$ csomópontja van és láthatjuk az 1 ábrán az élekhez rendelt költségekkel együtt. A lekérdezések száma $Q = 4$:

Az első lekérdezés esetében, ha $k = 4$, alkalmazhatjuk a feladatban leírt műveletet az 1–2 élre 3-szor, illetve az 1–4 él esetében egyszer, és $f(1, 4) = 11$. Ha $k = 5$, alkalmazhatjuk a feladatban leírt műveletet az 1–2 él esetében 3-szor, a 2–3 él esetében egyszer, illetve a 2–5 él esetében egyszer, tehát $f(1, 5) = 10$. Következésképpen az első lekérdezésre a válasz $f(1, 4) + f(1, 5) = 11 + 10 = 21$.

A második lekérdezés esetében csökkentjük az 1–2 él költségét egyszer, tehát $f(1, 1) = 16$.

A harmadik lekérdezés esetében észrevesszük, hogy az intervallum minden k értékére, $20 \leq k \leq 22$, a fa minden élének költsége 0-ra csökkenthető (és ennél több nincsen megengedve a művelet értelmezéséből adódólag), tehát az eredmény 0.

A negyedik lekérdezés esetében $k_1 = k_2 = 0$, tehát nem áll rendelkezésünkre egyetlen művelet sem amivel az élek költségeit csökkenthetnénk. Tehát a válasz minden 4-es csomópontból induló út költségének költségének az összege, vagyis $2 + 5 + 9 + 11 = 27$.

Feladat Detonator

Bemenet `detonator.in`
Kimenet `detonator.out`

Roman felügyelő egy robbanó szerkezet előtt áll, ami gyakorlatilag egy N szintű piramis, amelynek szintjeit 1-től N -ig számozzuk. Minden i szinten i bomba található, melyeket $B_{i,j}$ jelöl, ahol j az adott szinten található bomba sorszáma. Minden $B_{i,j}$ bomba esetén tudjuk, hogy a kezdeti időponttól számítva $T_{i,j}$ másodperc után bekövetkezik a robbanás. Roman feladata az, hogy hatékonyan hatástalanítsa a bombákat, a következő szabályok szerint:



1. Egy bomba hatástalanításának ideje 1 másodperc;
 2. Ahhoz, hogy hatástalaníthassuk a $B_{i,j}$ bombát, előbb hatástalanítani kell az alatta levő két bombát, amelyeken ez fekszik, vagyis $B_{i+1,j}$ és $B_{i+1,j+1}$. Az N -ik szinten levő bombák alatt nincsenek más bombák, tehát nem teljesítik ezt a szabályt.
1. ábra. Példa $N = 4 - re$. A $T_{i,j}$ értékek a bombákra vannak írva.

A szerkezet akkor lesz hatástalanítva, ha minden bomba hatástalanítva van. Roman nem szeretne kapkodni, ezért tudni szeretné, hogy mennyi a maximális X idő másodpercben mérve úgy, hogy ha a hatástalanítási művelet X másodpercet késik, a szerkezet még mindig hatástalanítható legyen. Más szóval mondva, mennyi X legnagyobb értéke, amelyre a hatástalanítás még megvalósítható, ha $T_{i,j}$ helyett $T_{i,j} - X$ értéket rakunk minden $1 \leq j \leq i \leq N$ esetben. Még az is előfordulhat, hogy X egy negatív szám, ha Roman túl későn érkezett a szerkezethez. Például, ha a szerkezet nem hatástalanítható az adott esetben, de ha Roman egy másodperccel hamarabb érkezett volna, akkor a hatástalanítás sikeres lenne, akkor $X = -1$. Ha a hatástalanítás egy másodperccel hamarabb nem sikerült volna, de két másodperccel hamarabb sikerült volna, akkor $X = -2$, és így tovább.

Követelmény

Adott Q teszt, mindegyikről ismerjük N -et és a $T_{i,j}$ értékeket minden $1 \leq j \leq i \leq N$ esetén, és ki kell számoljuk X értékét.

Bemeneti adatok

A `detonator.in` bemeneti állomány első sora tartalmazza Q -t, a tesztek számát. Következik a Q teszt leírása. Minden teszt esetében az első sor tartalmazza N értékét. A következő N sor mindegyike, 1-től N -ig számozva az adott szerkezet egy szintjét egyetlen sorban, természetes számokat, ahol a j -edik szám az i -edik sorból $T_{i,j}$. Vegyék észre, hogy az i -ik sor i elemet tartalmaz.

Kimeneti adatok

A `detonator.out` kimeneti állomány Q sort fog tartalmazni, minden teszt esetében az X számot, sorrendben.

Korlátok

- $1 \leq Q \leq 5$
- $1 \leq N \leq 1000$
- $1 \leq T_{i,j} \leq 10^9$, minden $1 \leq j \leq i \leq N$.

#	Pontszám	Korlátok
1	7	A $T_{i,j}$ értékek minden $1 \leq j \leq i \leq N$ esetén egyenlőek (minden bomba ugyanabban a pillanatban fog robbanni).
2	13	$N \leq 5, X \leq 10$
3	9	$N \leq 5$
4	14	$N \leq 50, X \leq 10$ si $T_{i,j} \geq \max\{T_{i+1,j}, T_{i+1,j+1}\}$ minden $1 \leq j \leq i < N$
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	Más megszorítások nincsenek.

Példák

detonator.in	detonator.out	Magyarázat
4	7	Az első tesztben $N = 2$, és van $N \cdot (N + 1)/2 = 3$ bombánk. Minden bomba 10 másodperc után robban. Tehát Roman legtöbb 7 másodpercet várhat, amikor még hatástalaníthatja a bombákat a 8-ik, 9-ik illetve 10-ik másodpercben. A második teszt az 1. ábrán látható. Ebben az esetben Roman azonnal el kell kezdje a bombák hatástalanítását, ugyanis a $B_{4,1}$ bomba egy másodperc után robban. Az egyetlen helyes hatástalanítási sorrend a bombák $B_{i,j}$ értéke pontosan ebben a sorrendben $1, 2, \dots, 10$. A harmadik teszt esetében létezik három bomba, amelyik azonnal robban, tehát Roman 2 másodperccel hamarabb kellett volna jöjjön, hogy sikeresen hatástalanítsa a szerkezetet.
2	0	
10	-2	
10 10	0	
4		
10		
7 9		
4 6 8		
1 3 2 5		
3		
9		
9 9		
1 1 1		
3		
6		
5 3		
4 4 4		

Feladat Zid

Bemenet `zid.in`
Kimenet `zid.out`

Andrát elbűvöli a kínai kultúra, de legfőképpen a Kínai Nagyfal, ezért elhatározta, hogy megépíti saját falát piros és sárga elemekből, melynek magassága N , szélessége pedig M .

Rendelkezésére áll akárhány elem melyeknek szélessége mindig 1, magassága pedig lehet akármennyi.

A piros elemek (hóng) magassága páratlan ($1, 3, 5, \dots$), míg a sárga elemek (huáng) magassága páros ($2, 4, 6, \dots$). **Az elemek nem forgathatók és csak függőlegesen helyezhetők.**



Tekintettel, hogy a kínai kultúrában a sárga szín a legértékesebb, Andra azt szeretné, hogy a falat alkotó sárga elemek összhosszának értéke legyen egyenlő K -val, aminek speciális szerencsehozó értéke van. Sőt, azt szeretné tudni, hogy **hány féle képpen** építheti fel a falat úgy, hogy ezt a tulajdonságot betartsa.

Tekintettel, hogy ez a követelmény túlságosan is „kínai” számára, (pedig beszéli a kínai nyelvet!), elhatározta, hogy segítségeket fogja kérni, mert azt hallotta, hogy a sárkány évében szerencsések vagytok.

Követelmény

Adottak az N , M és K értékek, határozzátok meg a különböző falak számát, amelyeket felépíthetünk az adott feltételek mellett.

Bemeneti adatok

A `zid.out` bemeneti állomány első sora tartalmazza az N , M és K értékeket, az előbb leírt jelennel.

Kimeneti adatok

A `zid.out` kimeneti állomány egyetlen sora tartalmazni fogja az adott feltételek mellett megépíthető különböző falak számát, **modulo** $10^9 + 7$.

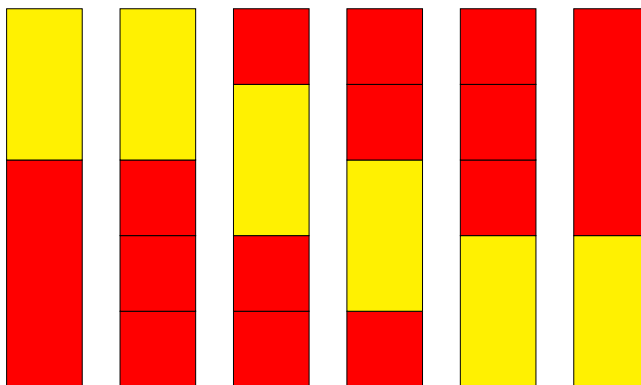
Korlátok

- $1 \leq N, M \leq 5\,000$
- $0 \leq K \leq N$
- Két falépítést azonosnak tekintünk, ha azonos típusú elemeket tartalmaz azonos pozíciókon.
Például, két féle képpen építhetünk 4×1 -es falat teljesen sárga színnel.

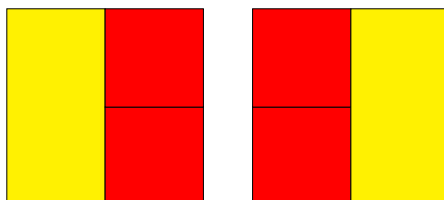
#	Pontszám	Korlátok
1	10	$N \leq 6, M = 1$
2	16	$N \leq 500, M = 1$
3	5	$N \leq 2\,500, M = 1, K = N$
4	6	$N \leq 2\,500, M = 1, K = 0$
5	14	$N \leq 2\,500, M = 1$
6	4	$N \leq 500, M = 2$
7	2	$N \leq 500, M = 3$
8	5	$N \leq 2\,500, M = 4$
9	17	$N \leq 2\,500, M \leq 10$
10	14	$N \leq 2\,500, M \leq 2\,500$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

Példák

zid.in	zid.out	Magyarázat
5 1 2	6	6 féle fal létezik, ahol magasság=5, szélesség=1 és $K = 2$. (Lásd az 1. ábrát.)
2 2 2	2	Van 2 falépítési lehetőség. (Lásd a 2. ábrát.)



1. ábra



2. ábra