

Feladat Dominoes

Bemenet dominoes.in
Kimenet dominoes.out

Adott egy 2 soros és n oszlopos mátrix, melyben k darab cella foglalt.

Adott q darab (x_1, y_1, x_2, y_2) alakú lekérdezés a következő jelentéssel: ha még el lesz foglalva **két különböző szabad cellája** a mátrixnak, az (x_1, y_1) és (x_2, y_2) , le lehet-e fedni a mátrixot teljesen 2×1 és 1×2 alakú dominókkal? Miután egy lekérdezést megválaszolunk, az ennek megfelelő foglalt cellák ismét felszabadulnak (a megfelelő változások nem maradnak meg a lekérdezések között).

Követelmény

Határozzuk meg minden lekérdezés esetén, hogy le lehet-e fedni a mátrixot teljesen 2×1 és 1×2 alakú dominókkal.

Bemeneti adatok

A dominoes.in bemeneti állomány első során az n és k értékek találhatók.

A következő k soron az elfoglalt pozíciók koordinátái találhatók (x, y) alakban.

A $(k+2)$ -ik soron a q érték szerepel. A következő q sor mindegyikén az x_1, y_1, x_2 și y_2 értékek találhatók szóközzel elválasztva, a feladatban megadott jelentéssel.

Kimeneti adatok

A dominoes.out kimeneti állomány q sort kell tartalmazzon, minden soron egy-egy lekérdezésnek megfelelő válasszal (1 ha le lehet fedni, illetve 0 ha nem lehet) abban a sorrendben, amelyben a kérdések a bemeneti állományban megtalálhatók.

Korlátok

- **Garantált, hogy mátrix kezdeti állapota teljesen lefedhető dominókkal.**
- $1 \leq x, x_1, x_2 \leq 2$
- $1 \leq y, y_1, y_2 \leq n$
- $0 \leq k \leq 100\,000$, k páros
- $1 \leq q \leq 100\,000$
- $1 \leq n \leq 1\,000\,000\,000$

#	Pontszám	Korlátok
1	9	$n, k, q \leq 8$
2	6	$k = 0$
3	11	$n, k, q \leq 5\,000$
4	12	$k, q \leq 5\,000$
5	23	$n \leq 100\,000$
6	6	$y_1 = y_2$ minden lekérdezés esetén
7	8	$y_1 \neq y_2$ minden lekérdezés esetén
8	25	nincsenek további megkötések

Példák

dominoes.in	dominoes.out	Magyarázat
5 2 1 1 2 5 4 1 3 2 3 2 1 1 5 2 3 2 4 1 2 2 4	0 1 1 1	Az alábbiakban megtalálható az egyes lekérdezések esetén a mátrix állapota a bemenetnek megfelelő sorrendben, a fekete négyzetek a foglalt cellákat ábrázolják:
6 4 1 1 2 3 1 4 2 6 3 1 2 2 2 1 2 1 3 1 2 1 5	0 1 0	A mátrix állapotai az egyes lekérdezések esetén:
6 2 1 1 2 3 3 1 4 2 6 1 4 2 5 1 3 2 5	1 0 0	A mátrix állapotai az egyes lekérdezések esetén:

Feladat Seqstr

Bemenet `seqstr.in`
Kimenet `seqstr.out`

Adott két sorozat, A és B , melyeknek minden eleme a $\{0, 1\}$ halmazból való.

Követelmény

- Határozzuk meg a B tömb azon egymástól különböző résztömbjeinek számát, melyek részsorozatai A -nak.
- Határozzuk meg egy megadott $B_{p...q}$ résztömb esetén az A azon részsorozatainak számát, amelyek ezzel azonosak.
- Határozzuk meg az A azon részsorozatainak számát, melyek résztömbjei B -nek.

Bemeneti adatok

A `seqstr.in` állomány első során az n érték található, mely az A sorozat hossza. A második soron található az A sorozat n darab eleme egy-egy szóközzel elválasztva. A harmadik soron található m , a B sorozat hossza. A negyedik soron található a B sorozat m darab eleme. Az ötödik soron a C szám található, mely a megválaszolendő kérdés sorszáma (1, 2 vagy 3).

Ha C értéke 2, akkor a hatodik soron a p és q számok találhatók egy szóközzel elválasztva.

Kimeneti adatok

A `seqstr.out` kimeneti állomány egyetlen sort kell tartalmazzon, melyre a kérdésnek megfelelő választ kell írni.

Korlátok

- Az eredmények kiszámítása és kiírása modulo $10^9 + 7$ kell történjen.
- Az A sorozat egy p hosszúságú részsorozatán egy

$$A_{t_1}, A_{t_2}, A_{t_3}, \dots, A_{t_p}$$

sorozatot értünk, melyet a

$$1 \leq t_1 < t_2 < t_3 < \dots < t_p \leq n$$

pozíciók határoznak meg.

- A B sorozat minden résztömbjét két,

$$1 \leq p \leq q \leq M$$

érték határozza meg és ez a résztömb

$$B_p, B_{p+1}, \dots, B_q.$$

A résztömb hossza a $q - p + 1$ értékkel egyenlő.

- A B sorozat két résztömbje, melyeket a p_1, q_1 és p_2, q_2 pozíciópárok határoznak meg, **különbözőnek** számít, ha különböző hosszúságúak, vagy ha egyforma hosszúak és létezik k úgy, hogy $p_1 \leq p_1 + k \leq q_1$ és $p_2 \leq p_2 + k \leq q_2$, illetve $B_{p_1+k} \neq B_{p_2+k}$.
- $1 \leq m \leq n$

#	Pontszám	Korlátok
1	7	$C = 1$ és $n \leq 20$
2	9	$C = 1$ és $21 \leq n \leq 500$
3	19	$C = 1$ és $501 \leq n \leq 5000$
4	3	$C = 2$ és $n \leq 20$
5	9	$C = 2$ és $21 \leq n \leq 100$
6	15	$C = 2$ és $101 \leq n \leq 5000$
7	9	$C = 3$ és $n \leq 20$
8	9	$C = 3$ és $21 \leq n \leq 100$
9	20	$C = 3$ és $101 \leq n \leq 500$

Példák

seqstr.in	seqstr.out	Magyarázat
5 1 1 0 0 1 3 0 1 1 1	4	Az első kérést kell megoldani. Azon különböző résztömbjei B-nek, amelyekre létezik A-ban részsorozat, a következők: 0, 1, 0 1 és 1 1. A B sorozat 0 1 1 résztömbjére nem létezik ezzel azonos részsorozat A-ban.
5 1 1 0 0 1 3 0 1 1 2 2 3	3	A 2. kérést kell megoldani. Azon A-beli részsorozatok, amelyek a B sorozat 1 1 résztömbjével egyenlők, a következő pozíciókon levő elemekből állnak: 1 2, 1 5 és 2 5.
5 1 1 0 0 1 3 0 1 1 3	10	A 3. kérést kell megoldani. Csak az 1, 2 és 3 hosszú részsorozatokat kell vizsgálni. Nincs az A-nak B-vel egyenlő 3 hosszú részsorozata, az 1 és 2 hosszúakból pedig azokat számoltuk meg, amelyek a 0, 1, 0 1 és 1 1 sorozatokkal azonosak.

Feladat Teze

Bemenet `teze.in`
Kimenet `teze.out`

Az informatikatanár m darab diák dolgozatát kell kijavítsa. A diákok n feladatot kellett megoldjanak a dolgozatban, melyek 1-től n -ig számozottak. Minden diák megoldotta az összes feladatot, tehát a tanár összesen $m \times n$ feladatot kell kijavítson. Észreveszi, hogy sorban kijavítva az első dolgozat feladatmegoldásait, majd a második dolgozat feladatmegoldásait és így tovább, nem mindig sikerül a lehető legrövidebb idő alatt befejezze a dolgozatok javítását. A tanár a következő módszert alkalmazza javításkor:

- A dolgozatok f darab fázisban mind ki lesznek javítva ($1 \leq f \leq n$).
- Minden i fázisban egy eddig még ki nem javított részhalmazát tekintjük a feladatoknak, S_i -t. Ezek a feladatok minden dolgozatban ki lesznek javítva mielőtt az elsőre visszatérnénk.
- Formálisan, megválasztunk egy $S_i \subseteq \{1, \dots, n\}$ részhalmazt úgy, hogy:
 - $S_i \cap S_j = \emptyset, \forall i, j \in \{1, \dots, f\}, i \neq j$
 - $S_1 \cup \dots \cup S_f = \{1, \dots, n\}$

Minden dolgozat javításának elkezdésekor be kell azonosítani a diák nevét, mely egy pontosan p másodpercet igénybe vevő folyamat minden alkalommal, akkor is, ha többször is visszatérünk ugyanarra a dolgozatra.

Egy dolgozat javításának megkezdése után, minden feladat megkeresése k másodpercet vesz igénybe. A megválasztott részhalmaz első feladatának javítása t_1 másodpercig tart, a második feladaté t_2 másodpercig stb. Garantált, hogy $t_1 < t_2 < \dots < t_n$. Minden alkalommal, amikor egy dolgozathoz visszatérünk és újra elkezdünk javítani abból egy másik részhalmaznak megfelelő részt, a megfelelő részhalmaz első feladatának javítása ismét t_1 időt vesz majd igénybe.

A t_1 érték a bementi fájlban meg lesz adva a $d_0, \dots, d_{q-1}$ értékekkel együtt, melyekből meghatározhatók a t sorozat további elemei a következő képlettel: $t_i = t_{i-1} + d_{i \bmod q}, \forall i \in \{2, \dots, n\}$, ahol $x \bmod y$ az x szám y -nal való osztási maradékát jelenti.

Követelmény

Határozzuk meg azt a minimális időt, amely alatt kijavítható az m darab dolgozat.

Bemeneti adatok

A `teze.in` bemeneti állomány első során az n, m, p és k számok találhatók egy-egy szóközzel elválasztva. A második soron adott két érték, t_1 és q , egy szóközzel elválasztva. A következő q soron a $d_0, \dots, d_{q-1}$ értékek találhatók.

Kimeneti adatok

A `teze.out` kimeneti állományba egyetlen számot kell írni, mely a javításhoz szükséges másodpercek száma $\bmod 10^9 + 7$.

Korlátok

- $1 \leq n \leq 1\,500\,000\,000$, ahol n a dolgozatokban megoldott feladatok száma
- $1 \leq m, q, k \leq 1\,000$, ahol m a diákok száma, q a sorozat hossza, illetve k jelképezi egy feladat megkereséséhez szükséges másodpercek számát
- $1 \leq p \leq 10^{10}$, ahol p azon másodpercek száma, melyekre egy diák nevének beazonosításához szükség van
- $1 \leq t_1, d_i \leq 10$

#	Pontszám	Korlátok
1	5	$n \leq 25$
2	10	$n \leq 50$
3	20	$n \leq 10\,000$
4	30	$n \leq 10^7$
5	7	$q = 1$
6	8	$q = 2$
7	20	nincsenek további megkötések

Példák

teze.in	teze.out	Magyarázat
2 10 5 2 1 1 2	130	Két feladat van a dolgozatban, továbbá $t_1 = 1$, $t_2 = t_1 + d_0 = 1 + 2 = 3$. Ha együtt javítjuk ki őket, minden dolgozat javítási ideje a következőképpen alakul: 5 másodperc a diák nevének beazonosítására, 2 másodperc az első feladat megkeresésére, 1 másodperc az első feladat javítására, 2 másodperc a második feladat megkeresésére és 3 másodperc a második feladat javítására, tehát összesen $5 + 2 + 1 + 2 + 3 = 13$ másodperc mind a 10 dolgozat esetén, tehát összesen 130 másodperc. Egy másik lehetőség az lenne, hogy kijavítjuk először az 1-es feladatot minden dolgozatban, majd ezt követően kijavítjuk a 2-es feladatot minden dolgozatban. Az első feladat javításához kell: 5 másodperc a javítás megkezdéséhez, 2 másodperc a feladat megkereséséhez és 1 másodperc a javításhoz, összesen $5 + 2 + 1 = 8$ másodperc dolgozatonként, ami összesen 80 másodperc az összes dolgozatra. A 2-es feladat javításánál az előzővel azonos számítást kell elvégezni, tehát összesen 160 másodpercre van szükség, tehát az első módszer hatékonyabb volt és annál jobb lehetőség nem létezik.