

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 részalmazát tekintjük a feladatoknak, S_i -t. Ezek a feladatok minden dolgozatban ki lesznek javítva mielőtt az elsőre visszatérnénk.
- Formálisan, megválasztunk egy $S_i \subseteq \{1, \dots, n\}$ részalmazt 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 részalmaz 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 részalmaznak megfelelő részt, a megfelelő részalmaz 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 mod $10^9 + 7$.

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 d 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	<p>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.</p> <p>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.</p>