

Problema Xidartros

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

A Radix Sort algoritmus, amelyet egy természetes számokat tartalmazó vektor rendezésére használunk, a következő lépések végrehajtását feltételezi:

- rendezzük a vektor elemeit az utolsó számjegy szerint, valamint egyenlőség esetén a vektorban lévő eredeti pozíció szerint;
- rendezzük a vektor elemeit az utolsó előtti számjegy szerint, valamint egyenlőség esetén az előző lépésben lévő pozíció szerint;
- ismételjük ezt a műveletet a legjelentősebb számjegyig (maximális helyiértéken szerepel) a sorozatban szereplő számok leírásában.

Például a 525, 417, 381, 291, 455 sorozat esetén az algoritmus a következő lépések elvégzését feltételezi:

- rendezzük az utolsó számjegy szerint, és a kapott sorozat: 381, 291, 525, 455, 417;
- rendezzük az utolsó előtti számjegy szerint, és a kapott sorozat: 417, 525, 455, 381, 291;
- rendezzük az első számjegy szerint, és a kapott sorozat: 291, 381, 417, 455, 525.

Ha az eredeti sorozatban az elemekhez az 1, 2, 3, 4, 5 indexeket rendeljük, és az algoritmus minden lépése után ezeket az indexeket az eredeti sorozatban az elemeknek megfelelő sorrend szerint írjuk, akkor minden lépés után a indexek sorrendje a következő:

- 3, 4, 1, 5, 2;
- 2, 1, 5, 3, 4;
- 4, 3, 2, 5, 1.

Észrevesszük, hogy ezen sorozatok értékei az eredeti vektor kezdeti indexeit tartalmazzák, amelyet rendezni szeretnénk. Így ezek minden lépésben 1-től N -ig a természetes számok egy permutációját alkotják.

Hasonlóan a sorozatot tetszőleges B számrendszer szerint rendezhetjük, ha átalakítjuk a számokat B számrendszerbe, majd alkalmazzuk ugyanazt az eljárást. Például ha a rendezendő sorozat 6, 8, 5 és alkalmazni akarjuk a Radix Sort algoritmust 2-es számrendszerben, az átalakított számok 110(6), 1000(8) és 101(5). A továbbiakban minden bit esetén a következő sorozatokat kapjuk:

- rendezzük az utolsó bit szerint: 110, 1000, 101;
- rendezzük az utolsó előtti bit szerint: 1000, 101, 110;
- rendezzük az utolsó előtti bit szerint: 1000, 101, 110;
- rendezzük a legjelentősebb bit szerint: 101, 110, 1000.

Következésképpen rendre a következő permutációkat kapjuk:

- 1, 2, 3;
- 2, 3, 1;
- 2, 3, 1;
- 3, 1, 2.

Követelmény

Pia cicának van egy N hosszúságú kezdeti sorozata, amelyre alkalmazta a Radix Sort algoritmust B alapú számrendszerben. Sajnos Mitzu testvére játszott Pia sorozatával, és a sorozat elveszett, de Pia-nak továbbra is megvannak a kapott permutációk, amelyeket az algoritmus meghívása során lépésenként kapott. Segítsetek Pia-nak hogy meghatározzon egy lehetséges eredeti sorozatot.

Bemeneti adatok

A bemeneti állomány első sorában a T szám található, amely a feladatban található tesztek száma. Minden teszt első sora 3 természetes számot tartalmaz: N - a sorozat elemeinek száma, B - a használt számrendszer alapja, amelyet a Radix Sort algoritmus meghívásai esetén használunk, és K - az algoritmus lépéseinek száma. A soron következő K sorban soronként N természetes szám található, amelyek az elemek eredeti indexeinek permutációi az algoritmus minden lépése után.

Kimeneti adatok

A kimeneti állományba T sor lesz kiírva, az i . sor az i . tesztre adott választ fogja tartalmazni. Ha a tesztnek van megoldása, akkor N természetes számot fog tartalmazni, amelyek az eredeti sorozat elemei (10-es számrendszerben írva) úgy, hogy a Radix Sort algoritmust alkalmazva a tesztben megadott alapú számrendszerben, az indexek permutációi az algoritmus minden lépése után ugyanaz legyen, mint amelyek a bemeneti állományban vannak megadva. Ha a tesztnek



nincs megoldása, akkor az i . sorban -1 lesz kiírva.

Restricții

- $1 \leq N, T \leq 10^6$
- $2 \leq B \leq 10^9$
- $1 \leq K \leq 64$
- Az input-ról az összes permutációból beolvasott elemek száma nem haladja meg 10^6 -t.
(Minden teszt esetén $N * K$ értéke $\leq 10^6$)
- Minden tesztre, amelynek van megoldása, létezik egy olyan megoldás, amely 0 és 10^{18} közötti elemeket tartalmaz, beleértve a határokat. **Minden olyan helyes megoldás elfogadható, amely ebből az intervallumból tartalmaz elemeket.**

#	Punctaj	Restricții
1	6	$N = 4, 1 \leq T, MAXVAL \leq 30$, minden tesztre a T teszt esetén. Ez esetben a $MAXVAL$ arra utal, hogy minden teszt esetén, amelynek van megoldása, létezik egy megoldás, amelyben a legnagyobb érték 30.
2	6	$N \leq B$, az összes T teszt esetén
3	11	$K = 1$, az összes T teszt esetén
4	13	$K = 2$, az összes T teszt esetén
5	17	$B = 2$, az összes T teszt esetén
6	19	$B = 10$, az összes T teszt esetén
7	28	Kezdeti megszorítások.

Exemple

xidartros.in	xidartros.out
3	525 417 381 291 455
5 10 3	6 8 5
3 4 1 5 2	-1
2 1 5 3 4	
4 3 2 5 1	
3 2 4	
1 2 3	
2 3 1	
2 3 1	
3 1 2	
3 2 1	
3 2 1	

Magyarázat

Az első 2 teszt esetén lásd a feladat kijelentésében szereplő magyarázatokat. A 3. teszt esetén nem létezik egyetlen sorozat sem, amelyet ha a Radix Sort algoritmussal rendezünk 2-es számrendszerben, és ha egyszer rendezünk, akkor A 3,2,1 permutációt kapjuk.