

Klany (A)

Limit pamięci: 512 MB

Limit czasu: 3.00 s

Bitek odnalazł drzewo genealogiczne swojej rodziny. Nie jest to zwykłe drzewo genealogiczne, ponieważ nie opisuje ono poszczególnych członków rodziny, ale klany należące do rodziny Bitka. Drzewo klanów składa się z N klanów, każdy klan bezpośrednio wywodzi się z innego klanu, poza klanem założycielskim. Każdy z klanów ma swój herb, jednakże Bitek zauważył, że pewne klany korzystały z tego samego herbu, niezależnie od odległości klanów w drzewie genealogicznym. Dla uproszczenia, klany opisane są kolejnymi liczbami naturalnymi $1, 2, \dots, N$, a rodzaj herbu i -tego klanu to h_i . Klan założycielski ma numer 1.

Bitek wymyślił następującą grę. Zaczyna od wyboru dwóch klanów u i v oraz rodzaju herbu h , a następnie szuka takiego klanu w , z którego wywodzą się klany u oraz v , oraz jego rodzaj herbu to h . Wśród wszystkich możliwych klanów w interesuje go taki, który jest najmłodszy (tzn. nie ma żadnego klanu w' , który spełniałby powyższe warunki oraz wywodziłby się z klanu w).

Formalnie mówimy, że klan u wywodzi się z klanu w , jeżeli zachodzi jeden z poniższych warunków:

- $u = w$,
- u wywodzi się bezpośrednio z w ,
- u wywodzi się bezpośrednio z pewnego klanu p , który wywodzi się z w .

Drzewo klanów rodziny Bitka okazało się na tyle duże, że zabawa sprawia mu większy kłopot, niż się początkowo spodziewał. Pomóż mu i napisz program, który będzie odpowiadał na zapytania Bitka.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby N oraz Q oznaczające odpowiednio liczbę klanów oraz liczbę zapytań Bitka. W następnym wierszu znajduje się opis drzewa, składający się z $N - 1$ liczb p_2, p_3, \dots, p_N . Wartość p_i mówi o tym z jakiego klanu bezpośrednio wywodzi się klan i . W następnym wierszu następuje N liczb h_1, \dots, h_N opisujących numery herbów kolejnych klanów. W następnych Q wierszach następuje opis zapytań. Każde zapytanie składa się z trzech liczb u, v, h wg opisu gry Bitka.

Wyjście

Należy wypisać Q wierszy odpowiadających na kolejne zapytania Bitka, będące numerami klanów, które są odpowiedzią na dane zapytanie. Jeżeli dla danego zapytania nie istnieje klan spełniający warunki Bitka, należy dla tego zapytania wypisać liczbę -1.

Ograniczenia

$1 \leq N, Q \leq 200\,000$, $1 \leq p_i, h_i, u, v, h \leq N$, $p_i < i$ dla $i = 2, \dots, N$.

Przykład

Wejście

```
5 3
1 1 3 3
1 3 1 2 1
2 5 1
4 3 3
2 2 3
```

Wyjście

```
1
-1
2
```

Wejście

Wyjście

10 10	1
1 2 2 3 5 5 6 6 8	-1
2 3 1 2 4 1 4 1 2 4	2
5 4 2	1
1 1 1	-1
10 6 3	2
10 9 2	-1
1 1 1	3
6 3 3	2
2 1 1	3
5 7 1	
3 5 3	
3 9 1	

Zoom (B)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Rolnik Bitek analizuje zdjęcia satelitarne swojego pola. Jego pole można reprezentować jako dwuwymiarowy układ współrzędnych. Z darmowych dostępnych internetowych źródeł udało mu się znaleźć N zdjęć, każde w kształcie prostokąta o dwóch naprzeciwległych wierzchołkach w punktach $x_{i,1}, y_{i,1}$ oraz $x_{i,2}, y_{i,2}$.

Bitkowi zależało na tym, żeby zdjęcia coraz lepiej przybliżyły konkretny obszar jego pola. Innymi słowy chciałby, żeby zdjęcia dało posegregować się w takiej kolejności, że każde następne zdjęcie jest pewną częścią poprzedniego. Formalnie, jedno zdjęcie jest pewną częścią innego zdjęcia, jeżeli prostokątny obszar zawarty na pierwszym zdjęciu w całości zawiera się w prostokątnym obszarze drugiego zdjęcia.

Niestety, Bitek nie jest pewien, czy takie ułożenie zdjęć jest w ogóle możliwe. Pomóż mu i napisz program, który odpowie na jego pytanie.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N oznaczająca liczbę znalezionych przez Bitka zdjęć. W następnych N wierszach następuje opis zdjęć. i -ty opis składa się z czterech liczb całkowitych $x_{i,1}, y_{i,1}$ oraz $x_{i,2}, y_{i,2}$, oznaczających pewne dwa naprzeciwległe wierzchołki prostokątnego obszaru pola Bitka zawartego na i -tym zdjęciu satelitarnym.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać słowo TAK, jeżeli zdjęcia da się posegregować wg wymagań Bitka, lub słowo NIE jeżeli nie jest to możliwe.

Ograniczenia

$1 \leq N \leq 200\,000$, $0 \leq x_{i,j}, y_{i,j} \leq 10^9$, $x_{i,1} \neq x_{i,2}$ oraz $y_{i,1} \neq y_{i,2}$ dla $i = 1, \dots, N$, $j = 1, 2$.

Przykład

Wejście

```
3
0 0 10 10
2 8 8 2
6 4 4 6
```

Wyjście

```
TAK
```

Wejście

```
2
0 0 10 10
1 1 11 11
```

Wyjście

```
NIE
```

Wyrównanie terenu (c)

Limit pamięci: 512 MB

Limit czasu: 4.00 s

Rolnik Bitek dysponuje również podłużną działką, którą chciałby zagospodarować. Chciałby rozpocząć od wyrównania poziomu działki. Działkę można podzielić na N fragmentów o wysokościach h_1, \dots, h_N . Bitek jest w stanie (jedną ze swoich rolniczych maszyn) wybrać pewien **spójny przedział** fragmentów tej samej wysokości i zwiększyć lub zmniejszyć ich wysokość o dokładnie 1. Przykładowo, jeżeli działka Bitka podzielona jest na fragmenty o wysokościach 1, 3, 5, 4, 3, Bitek mógłby zmniejszyć wysokość fragmentu 5 otrzymując wysokości 1, 3, 4, 4, 3, następnie zmniejszyć wysokość fragmentów o wysokości 4, otrzymując wysokości 1, 3, 3, 3, 3, a następnie zwiększyć pierwsze fragment dwukrotnie, otrzymując fragmenty 3, 3, 3, 3, 3.

Każda taka operacja jest okupiona dużym wysiłkiem rolnika Bitka, dlatego chciałby wykonać ich jak najmniej. Nie wie jednak, jaka powinna być ostateczna wysokość działki, dlatego zada Ci Q pytań. Twoim zadaniem jest napisać program, który dla każdego zapytania odpowie Bitkowi, jaką co najmniej liczbę przekształceń terenu będzie musiał dokonać, żeby wszystkie fragmenty działki miały określoną przez Bitka wysokość.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite N, Q , oznaczające długość działki Bitka oraz liczbę zapytań. W następnym wierszu następuje N liczb całkowitych h_1, \dots, h_N , oznaczających kolejne wysokości fragmentów działki. W następnych Q wierszach następuje opis zapytań. Każde zapytanie składa się z jednej liczby całkowitej z .

Wyjście

Dla każdego zapytania Bitka należy wypisać jedną liczbę całkowitą, oznaczającą minimalną liczbę przekształceń terenu, jaką musi wykonać Bitek, aby zrównać teren do wartości z zapytania.

Ograniczenia

$$1 \leq N, Q \leq 200\,000, 1 \leq h_i, z \leq 10^9.$$

Przykład

Wejście

```
5 3
7 1 1 4 6
9
5
6
```

Wyjście

```
8
7
6
```

Wejście

```
10 5
19 6 4 14 17 12 9 19 6 6
7
15
4
5
20
```

Wyjście

```
36
36
38
37
37
```

Ulubione liczby (D)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Algosia dostała na urodziny zestaw klocków w kształcie cyfr. Bardzo lubi układać z nich różne liczby, w szczególności upodobała sobie liczby A oraz B . Algosia chciałaby ułożyć liczby A lub B jak najwięcej razy z dostępnych klocków (nie ma znaczenia ile ułoży liczb A , a ile B , liczy się jedynie ile łącznie liczb udało jej się ułożyć). Algosia zaczęła się zastanawiać, ile razy jest w stanie je ułożyć? Algosia może użyć danego klocka do ułożenia tylko jednej liczby.

Wejście

W pierwszym wierszu dane są dwie liczby A oraz B . W następnym wierszu następuje 10 liczb c_0, \dots, c_9 , gdzie c_i oznacza liczbę klocków w kształcie cyfry i w zestawie Algosi.

Wyjście

W pierwszym i jedynym wierszu wyjścia należy wypisać jedną liczbę oznaczającą maksymalną możliwą liczbę liczb A lub B , które Algosia jest w stanie ułożyć ze swoich klocków.

Ograniczenia

$0 \leq A, B \leq 10^{18}, 0 \leq c_i \leq 100\,000$.

Przykład

Wejście

```
10 121
5 12 4 0 0 0 0 0 0 0
```

Wyjście

```
8
```

Wyjaśnienie

Algosia może ułożyć liczbę A pięciokrotnie oraz liczbę B trzykrotnie.

Wejście

```
139680958863427945 872771072295691019
27 29 20 8 23 30 8 11 40 21
```

Wyjście

```
22
```

Wyjaśnienie

Algosia może ułożyć liczbę A czterokrotnie oraz liczbę B jednokrotnie.

Redukcja (E)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Rozważmy następujący proces redukcji danej liczby naturalnej N :

- Jeżeli w zapisie binarnym liczby N w pewnym miejscu występują kolejno cyfry 101, to zmień te cyfry w 1.
- Jeżeli w zapisie binarnym liczby N w pewnym miejscu występują kolejno cyfry 010, to zmień te cyfry w 0.

Przykładowo, liczbę $5 = 101_{(2)}$ możemy w ten sposób zmienić w liczbę 1, a liczbę $90 = 1011010_{(2)}$ w $26 = 11010_{(2)}$ lub $22 = 10110_{(2)}$. Proces mogliśmy kontynuować na obu z tych liczb. Dla danej liczby N przez $R(N)$ oznaczmy zbiór wszystkich liczb, do których możemy zredukować liczbę N w dowolnej liczbie kroków. Przykładowo, $R(5) = \{5, 2\}$, $R(0) = \{0\}$ oraz $R(90) = \{90, 26, 22, 6\}$.

Twoim zadaniem jest, dla danej liczby K , znaleźć najmniejszą nieujemną liczbę całkowitą N_K taką, że $|R(N_K)| = K$. Innymi słowy, należy znaleźć najmniejszą liczbę naturalną taką, że można zredukować ją do dokładnie K liczb. Jako, że N może być bardzo duże, należy wypisać wartość reszty z dzielenia N_K przez 998 244 353.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita T oznaczająca liczbę zestawów testowych. W następnych T wierszach następuje opis każdego zestawu testowego, i -ty z nich składa się z jednej liczby całkowitej K_i .

Wyjście

Należy wypisać T wierszy. W i -tym wierszu należy wypisać jedną liczbę, resztę z dzielenia N_{K_i} przez 998 244 353.

Ograniczenia

$$1 \leq T \leq 20, 1 \leq K_i \leq 10^{12}.$$

Przykład

Wejście

3
2
6
1234567890

Wyjście

5
173
368322

Kość (F)

Limit pamięci: 512 MB

Limit czasu: 1.00 s

Algosia gra z rodzicami w grę planszową. Plansza składa się z N pól ponumerowanych kolejno $1, 2, \dots, N$. Pionek Algosi początkowo stoi na polu nr 1. Celem gry jest dotrzeć swoim pionkiem do ostatniego, N -tego pola. W każdej turze Algosia rzuca K -ścienną kością i przesuwa się o liczbę wylosowanych pól. Algosia zaczęła zastanawiać się, ile co najmniej rzutów kością musi wykonać, żeby dotrzeć do końca planszy?

Wejście

W pierwszym wierszu znajduje się jedna liczba naturalna T opisująca liczbę zestawów testowych. W następnych T wierszach znajduje się opis kolejnych zestawów testowych. Każdy opis składa się z dwóch liczb całkowitych N, K , oznaczających odpowiednio długość planszy oraz rozmiar kości.

Wyjście

Dla każdego zestawu testowego należy wypisać jedną liczbę całkowitą, oznaczającą minimalną liczbę rzutów kością konieczną do dotarcia do ostatniego pola planszy.

Ograniczenia

$1 \leq T \leq 1\,000, 1 \leq N, K \leq 10^9$.

Przykład

Wejście

3
17 5
5 1
6 14

Wyjście

4
4
1

Podział ciągu (G)

Limit pamięci: 512 MB

Limit czasu: 2.00 s

Algosia, jak każde dziecko w Bitocji, ma swój ulubiony ciąg N liczb a_1, a_2, \dots, a_N . Algosia lubi dzielić swój ciąg na spójne kawałki. Dla każdego kawałka określa jego *fajność* przez różnicę między największym a najmniejszym elementem danego kawałka. Przykładowo, gdyby ciąg $[2\ 6\ 4\ 8\ 3\ 4]$ podzielić na kawałki $[2\ 6\ 4]$ oraz $[8\ 3\ 4]$, to fajność tych kawałków wyniosłaby odpowiednio $6 - 2 = 4$ oraz $8 - 3 = 5$. Fajnością podziału ciągu określimy przez sumę fajności każdego z jego kawałków.

Algosia zastanawia się, jak mogłaby podzielić ciąg, żeby zmaksymalizować jego fajność. Ponadto, dokona Q zmian ciągu. Każda zmiana polega na dodaniu wartości x do wszystkich elementów a_l, a_{l+1}, \dots, a_r dla pewnych $l \leq r$. Napisz program, który po każdej zmianie ciągu odpowie jaka jest maksymalna fajność pewnego podziału tego ciągu.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite N oraz Q , oznaczające odpowiednio długość ciągu oraz liczbę zmian ciągu. W następnym wierszu znajduje się ciąg N liczb, oznaczające początkowe wartości ciągu. W następnych Q wierszach następują opisy kolejnych modyfikacji ciągu. Każda modyfikacja opisana jest przez trzy liczby całkowite l, r, x , oznaczające, że Algosia doda wartość x do wszystkich elementów od l -tego do r -tego włącznie.

Wyjście

Należy wypisać Q wierszy, każdy z nich opisujący największą możliwą fajność podziału ciągu po każdej modyfikacji.

Ograniczenia

$1 \leq N, Q \leq 200\,000$, $|a_i|, |x| \leq 10^8$, $1 \leq l \leq r \leq N$.

Przykład

Wejście

```
5 3
-1 0 1 2 3
2 3 1
2 2 4
1 2 -3
```

Wyjście

```
4
7
7
```

Wyjaśnienie

Możliwe najlepsze podziały po każdej modyfikacji to kolejno $[-1\ 1\ 2]$ $[2\ 3]$, $[-1\ 5]$ $[2]$ $[2\ 3]$, $[-4\ 2\ 2\ 2\ 3]$.