ARITMETICA MODULAR

Y EXPONENCIACIÓN BINARIA

Leo Tovíaz
Club de progra
CSC

A. Counting Orders

time limit per test: 1 second

memory limit per test: 256 megabytes

You are given two arrays a and b each consisting of n integers. All elements of a are pairwise distinct.

Find the number of ways to reorder a such that $a_i > b_i$ for all $1 \le i \le n$, modulo $10^9 + 7$.

Two ways of reordering are considered different if the resulting arrays are different.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the array a and b.

The second line of each test case contains n distinct integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the array a. It is guaranteed that all elements of a are pairwise distinct.

The second line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 10^9$) — the array b.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the number of ways to reorder array a such that $a_i > b_i$ for all $1 \le i \le n$, modulo $10^9 + 7$.

D. Dueling Digits

time limit per test: 2.5 seconds

memory limit per test: 256 megabytes

In the land of Numeria, two friends, Alice and Bob, are fascinated by numbers. Recently, they discovered a curious property about certain pairs of numbers and decided to explore it further. They are interested in finding pairs of numbers with the following properties:

1. Both numbers have N digits.
2. The sum of the digits of Alice's number is equal to the sum of the digits of Bob's number.
3. For any digit position i, the i-th digit of Alice's number is different from the i-th digit of Bob's number.
4. Both numbers cannot start with the digit zero.

You have Q queries, and for each query, you need to determine how many pairs of numbers exist that satisfy these conditions for a given number length N.

Input

The first line contains an integer Q $(1 \leq Q \leq 800)$, the number of queries.

Each of the next Q lines contains a single integer N $(1 \leq N \leq 800)$, representing the length of the numbers.

Output

For each query, print a single integer representing the number of valid pairs of numbers that satisfy the conditions for the given length N, because this number can be very large print it modulo $10^9 + 7$.

F. Finding the Best Guess

time limit per test: 5 seconds

memory limit per test: 256 megabytes

Huron Casino is running a strange gambling game on trees. The game consists on guessing the final value of the number S generated for the algorithm described below:

Given a tree with n nodes, where each node has an integer a_i written on it, and an integer S initially equal to 0. The algorithm consists on repeating the following steps n times:

1. Choose a random node u that has not been removed yet. All nodes that have not been removed have the same probability of been chosen.
2. Add a_v to S for all the nodes v, such that there is a path from u to v passing through nodes and edges that have not been removed yet.
3. Remove u and all the edges incident to it.

The game is played for a certain number of participants. The player (or players) whose guess is closer to the final value of S is the winner of the game.

You are an expert on greedy algorithms, for that reason you think that betting for the expected value of S is the best strategy. Write a program that finds the expected value of S.

Input

The first line contain an integer n ($1 \le n \le 10^5$).

The second line contains n integers a_i ($1 \le a_i \le 10^5$).

Each of the following $n - 1$ lines contains two integers u and v ($1 \le u, v \le n$) indicating that there is an edge connecting nodes u and v.

Output

It can be shown that the expected value of S can be expressed as an irreducible fraction P/Q such that P and Q are non-negative integers. Print $P \cdot Q^{-1}$ modulo 998244353.

A. Counting Orders

time limit per test: 1 second

memory limit per test: 256 megabytes

You are given two arrays a and b each consisting of n integers. All elements of a are pairwise distinct.

Find the number of ways to reorder a such that $a_i > b_i$ for all $1 \le i \le n$, modulo $10^9 + 7$.

Two ways of reordering are considered different if the resulting arrays are different.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the array a and b.

The second line of each test case contains n distinct integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the array a. It is guaranteed that all elements of a are pairwise distinct.

The second line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 10^9$) — the array b.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the number of ways to reorder array a such that $a_i > b_i$ for all $1 \le i \le n$, modulo $10^9 + 7$.

D. Dueling Digits

time limit per test: 2.5 seconds

memory limit per test: 256 megabytes

In the land of Numeria, two friends, Alice and Bob, are fascinated by numbers. Recently, they discovered a curious property about certain pairs of numbers and decided to explore it further. They are interested in finding pairs of numbers with the following properties:

1. Both numbers have N digits.
2. The sum of the digits of Alice's number is equal to the sum of the digits of Bob's number.
3. For any digit position i, the i-th digit of Alice's number is different from the i-th digit of Bob's number.
4. Both numbers cannot start with the digit zero.

You have Q queries, and for each query, you need to determine how many pairs of numbers exist that satisfy these conditions for a given number length N.

Input

The first line contains an integer Q $(1 \leq Q \leq 800)$, the number of queries.

Each of the next Q lines contains a single integer N $(1 \leq N \leq 800)$, representing the length of the numbers.

Output

For each query, print a single integer representing the number of valid pairs of numbers that satisfy the conditions for the given length N, because this number can be very large print it modulo $10^9 + 7$.

F. Finding the Best Guess

time limit per test: 5 seconds

memory limit per test: 256 megabytes

Huron Casino is running a strange gambling game on trees. The game consists on guessing the final value of the number S generated for the algorithm described below:

Given a tree with n nodes, where each node has an integer a_i written on it, and an integer S initially equal to 0. The algorithm consists on repeating the following steps n times:

1. Choose a random node u that has not been removed yet. All nodes that have not been removed have the same probability of been chosen.
2. Add a_v to S for all the nodes v, such that there is a path from u to v passing through nodes and edges that have not been removed yet.
3. Remove u and all the edges incident to it.

The game is played for a certain number of participants. The player (or players) whose guess is closer to the final value of S is the winner of the game.

You are an expert on greedy algorithms, for that reason you think that betting for the expected value of S is the best strategy. Write a program that finds the expected value of S.

Input

The first line contain an integer n $(1 \leq n \leq 10^5)$.

The second line contains n integers a_i $(1 \leq a_i \leq 10^5)$.

Each of the following $n - 1$ lines contains two integers u and v $(1 \leq u, v \leq n)$ indicating that there is an edge connecting nodes u and v.

Output

It can be shown that the expected value of S can be expressed as an irreducible fraction P/Q such that P and Q are non-negative integers. Print $P \cdot Q^{-1}$ modulo 998244353.

PROPIEDADES BÁSICAS

Suma

(a + b) mod m = (a mod m) + (b mod m) mod m

Resta

(a – b) mod m = (a mod m) – (b mod m) mod m

Multiplicación

(a * b) mod m = (a mod m) * (b mod m) mod m

Potencia

(a ^ b) mod m = (a mod m)^b mod m

DIVISIÓN

$$(a / b) \bmod m = a * b^{(-1)} \bmod m$$

El b^(-1) no es simplemente una potencia negativa "normal", por el hecho de estar trabajando con módulos ya no podemos aplicar la propeidad algebraica donde "pasamos" b hacia abajo.

INVERSO MULTIPLICATIVO,

INVERSO MULTIPLICATIVO MODULAR

El inverso multiplicativo de a es un número b que al multiplicarlo por a, el resultado es 1: b=5, a=1/5.

El inverso multiplicativo modular de a modulo m es un número b que al multiplicarlo por a, el resultado es congruente modulo 1.

a=3

b=4

m=11

CONGRUENCIA MODULO M

expr1 ≅ expr2 (mod m)
expr1 mod m = expr2 mod m

“Dos números son congruentes módulo m si tienen el mismo módulo”...
4 ≅ 16 (mod 12)

*El congruente modulo m de a solo existe si a y m son coprimos

INVERSO MULTIPLICATIVO MODULAR

El inverso multiplicativo modular de a modulo m es un número b que al multiplicarlo por a, el resultado es congruente modulo 1.

a=3

m=11

b=?...4

*a y m deben ser coprimos para que exista en inverso multiplicativo modular.

Entonces nomas es buscar un numerillo que satisfaga esa formulilla?

Para eso hay 2 métodos, veremos el método de "pequeño teorema de Fermat".

PEQUEÑO TEOREMA DE FERMAT

Si m es un número primo y a no es divisible por m, la siguiente fórmula aplica:

a^(m-1) ≅ 1 (mod m)

INVERSO CON EL TEOREMA

Entonces, recordando que para el inverso multiplicativo modular ocupamos un número b para multiplicarlo por a y que nos de 1 tras aplicarle el módulo indicado.. Podemos descomponer un poco la fórmula del teorema para generar el número b: $a*a^{(m-2)} \cong 1 \pmod{m}$, de aqui concluimos que necesitamos $a^{(m-2)} \bmod m$

P/a=5,m=3...

$b=5^{(3-2)} \pmod{3}$

$b=5 \bmod 3 = 2$

$5*2 = 10$, $10 \bmod 3 = 1$

Aunque multiplicar a por m - 2 veces puede hacer que truene el programa (si el módulo es algo como 10^9 + 7 o 998_244_353). De aquí surge una implementación llamada exponentiation by squaring/binary exponentiation:
*Esta función se puede usar para calcular x^n mod m eficientemente, puede ser el teorema de Fermat o no.

```java
private static long pow_mod(long x, long n, long m) {
  long res = 1;
  while (n > 0) {
    if ((n & 1) == 1) {
      res = (res * x) % m;
    }
    n >>= 1;
    x = (x * x) % m;
  }
  return res;
}
```

CALCULO DE INV DESGLOSADO

<u>10/3 mod 7 = 10 * 3^(-1) mod 7</u>

1. Primero calcular inverso/inverso multiplicativo modular de b=3 modulo=7.

*Como 3 y 7 son coprimos, el inverso existe.

*Como 7 es primo, podemos usar Fermat.

2. Desarrollo de Fermat (esto lo hace la función pow_mod):

x = a^(m-2) mod m

x = 3^5 mod 7

x = 243 mod 7

x = 5

... 3*5 = 15

... 15 mod 7 = 1

3. Reemplazar inverso de b en nuestra propiedad de división:

10/3 mod 7 = 10 * 5 mod 7

Solo nos queda confiar que nuestra "conversión" o "transformación" es correcta y es otra forma de representar una división en la aritmética modular.

MAS POSIBILIDADES

En teoria cualquier problema en el que se deba hacer una división y haya que obtener su módulo como respuesta, se puede aplicar el inverso multiplicativo modular.

Por ejemplo, veamos como cambia la implementación de binomio coeficiente...

SIN MOD // CON MOD

```
ll bincoe(ll n, ll k) {
  if (k > n - k) k = n - k;  // Optimización: C(n, k) == C(n, n-k)

  ll res = 1;
  for (int i = 0; i < k; i++) {
    res *= (n - i);
    res /= (i + 1);
  }
  return res;
}
```

```
private static long pow_mod(long x, long n, long m) {
  long res = 1;
  while (n > 0) {
    if ((n & 1) == 1) {
      res = (res * x) % m;
    }
    n >>= 1;
    x = (x * x) % m;
  }
  return res;
}
private static long bincoe(int n, int k) {
  if (k == 0 || k == n) return 1;

  long res = 1;
  for (int i = 0; i < k; i++) {
    res = (res * ((n - i) % MOD)) % MOD;
    res = (res * pow_mod(i + 1, MOD - 2, MOD)) % MOD;
  }
  return res;
}
```

PROBLEMAS

1. **CF 1178C - Tiles (PFC)**
2. **CF 1475E - Advertising Agency (Bincoe)**
3. **Vjudge https://vjudge.net/problem/EOlymp-9606 - Modular Division (inverso multiplicativo modular)**

Estudio
Shonos

MUCHAS GRACIAS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vulputate nulla at ante rhoncus, vel efficitur felis condimentum. Proin odio odio.

911-234-5678 www.unsitiogenial.es