Expected Damage

time limit per test

2 seconds

memory limit per test

256 megabytes

input

standard input

output

standard output

You are playing a computer game. In this game, you have to fight nn monsters.

To defend from monsters, you need a shield. Each shield has two parameters: its current durability aa and its defence rating bb. Each monster has only one parameter: its strength dd.

When you fight a monster with strength dd while having a shield with current durability aa and defence bb, there are three possible outcomes:

* if a=0a=0, then you receive dd damage;
* if a>0a>0 and d≥bd≥b, you receive no damage, but the current durability of the shield decreases by 11;
* if a>0a>0 and d<bd<b, nothing happens.

The ii-th monster has strength didi, and you will fight each of the monsters exactly once, in some random order (all n!n! orders are equiprobable). You have to consider mm different shields, the ii-th shield has initial durability aiai and defence rating bibi. For each shield, calculate the expected amount of damage you will receive if you take this shield and fight the given nn monsters in random order.

**Input**

The first line contains two integers nn and mm (1≤n,m≤2⋅1051≤n,m≤2⋅105) — the number of monsters and the number of shields, respectively.

The second line contains nn integers d1d1, d2d2, ..., dndn (1≤di≤1091≤di≤109), where didi is the strength of the ii-th monster.

Then mm lines follow, the ii-th of them contains two integers aiai and bibi (1≤ai≤n1≤ai≤n; 1≤bi≤1091≤bi≤109) — the description of the ii-th shield.

**Output**

Print mm integers, where the ii-th integer represents the expected damage you receive with the ii-th shield as follows: it can be proven that, for each shield, the expected damage is an irreducible fraction xyxy, where yy is coprime with 998244353998244353. You have to print the value of x⋅y−1mod998244353x⋅y−1mod998244353, where y−1y−1 is the inverse element for yy (y⋅y−1mod998244353=1y⋅y−1mod998244353=1).

**Examples**

**input**

**Copy**

3 2

1 3 1

2 1

1 2

**output**

**Copy**

665496237

1

**input**

**Copy**

3 3

4 2 6

3 1

1 2

2 3

**output**

**Copy**

0

8

665496236