

B. AND Sequences

A sequence of n non-negative integers ($n \geq 2$) a_1, a_2, \dots, a_n is called good if for all i from 1 to $n - 1$ the following condition holds true:

$$a_1 \& a_2 \& \dots \& a_i = a_{i+1} \& a_{i+2} \& \dots \& a_n,$$

where $\&$ denotes the [bitwise AND operation](#).

You are given an array a of size n ($n \geq 2$). Find the number of permutations p of numbers ranging from 1 to n , for which the sequence $a_{p_1}, a_{p_2}, \dots, a_{p_n}$ is good. Since this number can be large, output it modulo $10^9 + 7$.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$), denoting the number of test cases.

The first line of each test case contains a single integer n ($2 \leq n \leq 2 \cdot 10^5$) — the size of the array.

The second line of each test case contains n integers a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — the elements of the array.

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

Output

Output t lines, where the i -th line contains the number of good permutations in the i -th test case modulo $10^9 + 7$.

array of vector<int>arr(n) where element :

Index	0	1	2	3	4
Value	0	2	0	3	0

```
// find minimum value
int mn = *vec_min(arr);

// cnt minimum value
int cnt = 0;

for (auto val : arr)
{
```

```

if (mn == val)

    cnt++;

// if minimum value != mn & value --> print 0 becuae no good permutation

if ((mn & val) != mn)

{

    cout << 0 << endl;

    return;

}

}

// permutation cnt

/*

index 0 1 2 3 4 5 6 7 8 9 10 11 12 13 ..... n-1

if array mn X X X X X X X X X x  X X ..... mn this permutaiton is good

so , (n-2)P(n-2) = (n-2)! and minimum count take first index cnt P 1 and last index (cnt-1) P 1
*/

```

Index	0	1	2	3	4
Value	0	2	0	3	0

```

int ans = (1LL * fact(n - 2) * (1LL * cnt * (cnt - 1) % mod)) % mod;
cout << ans << endl;

```