A. Stickogon

```
Input file:     standard input
Output file:    standard output
Time limit:     1 second
Memory limit:   256 megabytes
```

You are given n sticks of lengths $a_1, a_2, \ldots, a_n$. Find the maximum number of regular (equal-sided) polygons you can construct simultaneously, such that:

- Each side of a polygon is formed by exactly one stick.
- No stick is used in more than 1 polygon.

Note: Sticks cannot be broken.

Input

The first line contains a single integer t $(1 \le t \le 100)$ — the number of test cases.

The first line of each test case contains a single integer n $(1 \le n \le 100)$ — the number of sticks available.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 100)$ — the stick lengths.

Output

For each test case, output a single integer on a new line — the maximum number of regular (equal-sided) polygons you can make simultaneously from the sticks available.

Standard Input	Standard Output

4
1
1
2
1 1
6
2 2 3 3 3 3
9
4 2 2 2 2 4 2 4 4
	0
	0
	1
	2

Note

In the first test case, we only have one stick, hence we can't form any polygon.

In the second test case, the two sticks aren't enough to form a polygon either.

In the third test case, we can use the 4 sticks of length 3 to create a square.

In the fourth test case, we can make a pentagon with side length 2, and a square of side length 4.

,

B. A BIT of a Construction

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Given integers n and k, construct a sequence of n non-negative (i.e. ≥ 0) integers $a_1, a_2, \ldots, a_n$ such that

1. $\sum\limits_{i=1}^{n} a_i = k$
2. The **number** of 1s in the binary representation of $a_1 | a_2 | \ldots | a_n$ is maximized, where $|$ denotes the bitwise OR operation.

Input

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

The only line of each test case contains two integers n and k ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq k \leq 10^9$) — the number of non-negative integers to be printed and the sum respectively.

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 a sequence $a_1, a_2, \ldots, a_n$ on a new line that satisfies the conditions given above.

If there are multiple solutions, print any of them.

Standard Input	Standard Output

4
1 5
2 3
2 5
6 51
	5
	1 2
	5 0
	3 1 1 32 2 12

Note

In the first test case, we have to print exactly one integer, hence we can only output 5 as the answer.

In the second test case, we output $1, 2$ which sum up to 3, and $1|2 = (11)_2$ has two 1s in its binary representation, which is the maximum we can achieve in these constraints.

In the fourth test case, we output $3, 1, 1, 32, 2, 12$ which sum up to 51, and $3|1|1|32|2|12 = (101\,111)_2$ has five 1s in its binary representation, which is the maximum we can achieve in these constraints.

,

C. How Does the Rook Move?

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given an $n \times n$ chessboard where you and the computer take turns alternatingly to place white rooks & black rooks on the board respectively. While placing rooks, you have to ensure that no two rooks attack each other. Two rooks attack each other if they share the same row or column **regardless of color**.

A valid move is placing a rook on a position (r, c) such that it doesn't attack any other rook.

You start first, and when you make a valid move in your turn, placing a white rook at position (r, c), the computer will mirror you and place a black rook at position (c, r) in its turn. If $r = c$, then the computer can't mirror your move, and skips its turn.

You have already played k moves with the computer (the computer tries to mirror these moves too), and you must continue playing the game until there are no valid moves remaining. How many different final configurations are possible when you continue the game after the k moves? It is guaranteed that the k moves and the implied computer moves are valid. Since the answer may be large, print it modulo $10^9 + 7$.

Two configurations are considered different if there exists a coordinate (r, c) which has a rook in one configuration, but not in the other **or** the color of the rook on the coordinate is different.

Input

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

The first line of each test case contains two integers n and k $(1 \le n \le 3 \cdot 10^5, 0 \le k \le n)$ — the size of the chessboard and the number of moves you have already played respectively.

Each of the next k lines of the test case contains two integers r_i and c_i, denoting the i-th move you made.

It is guaranteed that the k moves and the implied computer moves are valid.

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, output a single integer on a new line — the total number of possible final configurations modulo $10^9 + 7$.

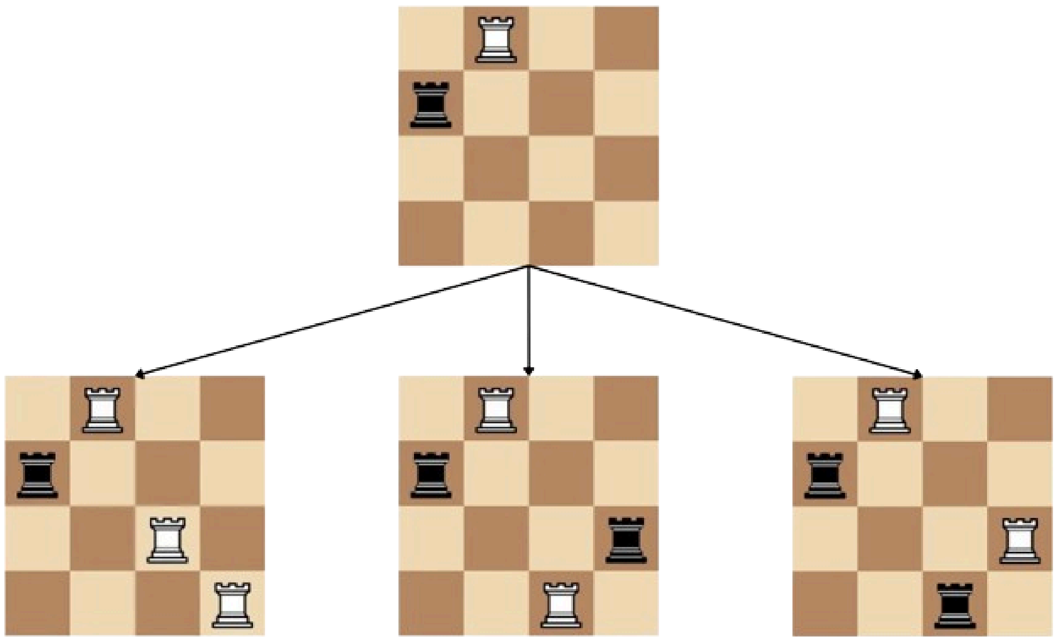
Standard Input	Standard Output

```
3
4 1
1 2
8 1
7 6
1000 4
4 4
952 343
222 333
90 91
```
	```
	3
	331
	671968183
	```

Note

In the first test case, we have a 4×4 grid and you've already played 1 move. After you and the computer play a turn, we have a white rook at $(1, 2)$, and a black rook at $(2, 1)$. There are three possible configurations from this state —

1. You place a white rook at $(3, 4)$ and the computer places a black rook at $(4, 3)$ as a response.
2. You place a white rook at $(4, 3)$ and the computer places a black rook at $(3, 4)$ as a response.
3. You place a white rook at $(3, 3)$ and then at $(4, 4)$, or the other way around. They both result in the same configuration.



,

D. A BIT of an Inequality

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given an array $a_1, a_2, \ldots, a_n$. Find the number of tuples (x, y, z) such that:

- $1 \le x \le y \le z \le n$, and
- $f(x, y) \oplus f(y, z) > f(x, z)$.

We define $f(l, r) = a_l \oplus a_{l+1} \oplus \ldots \oplus a_r$, where $\oplus$ denotes the bitwise XOR operation.

Input

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

The first line of each test case contains a single integer n $(1 \le n \le 10^5)$.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$.

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

Output

For each test case, output a single integer on a new line — the number of described tuples.

Standard Input	Standard Output

3
3
6 2 4
1
3
5
7 3 7 2 1
	4
	0
	16

Note

In the first case, there are 4 such tuples in the array $[6, 2, 4]$:

- $(1, 2, 2)$: $(a_1 \oplus a_2) \oplus (a_2) = 4 \oplus 2 > (a_1 \oplus a_2) = 4$
- $(1, 1, 3)$: $(a_1) \oplus (a_1 \oplus a_2 \oplus a_3) = 6 \oplus 0 > (a_1 \oplus a_2 \oplus a_3) = 0$
- $(1, 2, 3)$: $(a_1 \oplus a_2) \oplus (a_2 \oplus a_3) = 4 \oplus 6 > (a_1 \oplus a_2 \oplus a_3) = 0$
- $(1, 3, 3)$: $(a_1 \oplus a_2 \oplus a_3) \oplus (a_3) = 0 \oplus 4 > (a_1 \oplus a_2 \oplus a_3) = 0$

In the second test case, there are no such tuples.

,

E. Carousel of Combinations

```
Input file:     standard input
Output file:    standard output
Time limit:     2.5 seconds
Memory limit:   256 megabytes
```

You are given an integer n. The function $C(i,k)$ represents the number of distinct ways you can select k distinct numbers from the set $\{1,2,\ldots,i\}$ and arrange them in a circle[†].

Find the value of

$$\sum_{i=1}^{n}\sum_{j=1}^{i}\left(C(i,j) \bmod j\right).$$

Here, the operation $x \bmod y$ denotes the remainder from dividing x by y.

Since this value can be very large, find it modulo 10^9+7.

[†] In a circular arrangement, sequences are considered identical if one can be rotated to match the other. For instance, $[1,2,3]$ and $[2,3,1]$ are equivalent in a circle.

Input

The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

The only line of each test case contains a single integer n ($1 \le n \le 10^6$).

Output

For each test case, output a single integer on a new line — the value of the expression to be calculated modulo 10^9+7.

Standard Input	Standard Output
4 1 3 6 314159	0 4 24 78926217

Note

In the first test case, $C(1,1) \bmod 1 = 0$.

In the second test case:

- $C(1,1)=1$ (the arrangements are: $[1]$);
- $C(2,1)=2$ (the arrangements are: $[1]$, $[2]$);
- $C(2,2)=1$ (the arrangements are: $[1,2]$);
- $C(3,1)=3$ (the arrangements are: $[1]$, $[2]$, $[3]$);
- $C(3,2)=3$ (the arrangements are: $[1,2]$, $[2,3]$, $[3,1]$);
- $C(3,3)=2$ (the arrangements are: $[1,2,3]$, $[1,3,2]$).

In total,
$(C(1,1) \bmod 1)+(C(2,1) \bmod 1)+(C(2,2) \bmod 2)+(C(3,1) \bmod 1)+(C(3,2) \bmod 2)+(C(3,3) \bmod 3)=4$.

,

F1. Frequency Mismatch (Easy Version)

```
Input file:     standard input
Output file:    standard output
Time limit:     4 seconds
Memory limit:   512 megabytes
```

This is the easy version of the problem. The difference between the two versions of this problem is the constraint on k. You can make hacks only if all versions of the problem are solved.

You are given an undirected tree of n nodes. Each node v has a value a_v written on it. You have to answer queries related to the tree.

You are given q queries. In each query, you are given 5 integers, u_1, v_1, u_2, v_2, k. Denote the count of nodes with value c on path $u_1 \to v_1$ with x_c, and the count of nodes with value c on path $u_2 \to v_2$ with y_c. If there are z such values of c such that $x_c \neq y_c$, output any $\min(z, k)$ such values in any order.

Input

The first line contains one integer n ($1 \leq n \leq 10^5$) — the number of nodes in the tree.

The next line contains n integers, $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^5$) — the value written on each node of the tree.

Then $n - 1$ lines follow. Each line contains two integers u and v ($1 \leq u, v \leq n, u \neq v$) denoting an edge of the tree. It is guaranteed that the given edges form a tree.

The next line contains one integer q ($1 \leq q \leq 10^5$) — the number of queries.

Then q lines follow. Each line contains five integers u_1, v_1, u_2, v_2, k ($1 \leq u_1, v_1, u_2, v_2 \leq n$, $k = 1$).

Output

For each query, output on a separate line. For a query, first output $\min(z, k)$ and then on the same line, output any $\min(z, k)$ values in any order which occur a different number of times in each path.

Standard Input	Standard Output

5
5 2 3 4 3
1 2
1 3
2 4
2 5
3
1 4 4 5 1
2 3 2 3 1
5 5 4 3 1
	1 5
	0
	1 2

Note

For query 1, the first path is $1 \to 2 \to 4$, coming across the multiset of values $\{5, 2, 4\}$. On the second path $4 \to 2 \to 5$, we have the multiset $\{4, 2, 3\}$. Two numbers — 3 and 5 occur a different number of times, hence we print one of them.

In query 2, there is no difference between the paths, hence we output 0.

In query 3, the first path is just the node 5, resulting in the multiset $\{3\}$, and the second path $4 \to 2 \to 1 \to 3$ gives $\{4, 2, 5, 3\}$. The numbers 5, 2 and 4 occur a different number of times.

F2. Frequency Mismatch (Hard Version)

```
Input file:     standard input
Output file:    standard output
Time limit:     4.5 seconds
Memory limit:   512 megabytes
```

This is the hard version of the problem. The difference between the two versions of this problem is the constraint on k. You can make hacks only if all versions of the problem are solved.

You are given an undirected tree of n nodes. Each node v has a value a_v written on it. You have to answer queries related to the tree.

You are given q queries. In each query, you are given 5 integers, u_1, v_1, u_2, v_2, k. Denote the count of nodes with value c on path $u_1 \rightarrow v_1$ with x_c, and the count of nodes with value c on path $u_2 \rightarrow v_2$ with y_c. If there are z such values of c such that $x_c \neq y_c$, output any $\min(z, k)$ such values in any order.

Input

The first line contains one integer n ($1 \leq n \leq 10^5$) — the number of nodes in the tree.

The next line contains n integers, $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^5$) — the value written on each node of the tree.

Then $n - 1$ lines follow. Each line contains two integers u and v ($1 \leq u, v \leq n, u \neq v$) denoting an edge of the tree. It is guaranteed that the given edges form a tree.

The next line contains one integer q ($1 \leq q \leq 10^5$) — the number of queries.

Then q lines follow. Each line contains five integers u_1, v_1, u_2, v_2, k ($1 \leq u_1, v_1, u_2, v_2 \leq n$, $1 \leq k \leq 10$).

Output

For each query, output on a separate line. For a query, first output $\min(z, k)$ and then on the same line, output any $\min(z, k)$ values in any order which occur a different number of times in each path.

Standard Input	Standard Output
5	2 3 5
5 2 3 4 3	0
1 2	1 5
1 3	3 5 2 4
2 4	
2 5	
4	
1 4 4 5 3	
2 3 2 3 1	
1 4 4 5 1	
5 5 4 3 10	

Note

For query 1, the first path is $1 \rightarrow 2 \rightarrow 4$, coming across the multiset of values $\{5, 2, 4\}$. On the second path $4 \rightarrow 2 \rightarrow 5$, we have the multiset $\{4, 2, 3\}$. Two numbers — 3 and 5 occur a different number of times, hence we print them both.

In query 2, there is no difference between the paths, hence we output 0.

In query 3, we have the same paths as query 1, but we need to output only 1 value, hence we output 5.

In query 4, the first path is just the node 5, resulting in the multiset $\{3\}$, and the second path $4 \rightarrow 2 \rightarrow 1 \rightarrow 3$ gives $\{4, 2, 5, 3\}$. The numbers 5, 2 and 4 occur a different number of times.