Problem A. Rikka with Minimum Spanning Trees

Input file: standard input
Output file: standard output

Hello everyone! I am your old friend Rikka. Welcome to Xuzhou. This is the first problem, which is a problem about the minimum spanning tree (MST). I promise you all that this should be the **easiest problem** for most people.

A minimum spanning tree, or minimum weight spanning tree, is a subset of edges from an edge-weighted undirected graph, which forms a tree with the minimum possible total edge weight that connects all the vertices together without any cycles.

In this problem, Rikka wants you to calculate the summation of total edge weights through all MSTs for a given graph, which obviously equals to the product of the total edge weight in an MST and the total number of different MSTs. Note that two spanning trees are different if the sets of their edges are different. In addition, a disconnected graph could have no MSTs, the number of whose different MSTs is zero.

To decrease the size of the input, Rikka provides an edge-weighted undirected graph via a random number generator with given random seeds, denoted by two integers k_1 and k_2. Supposing the number of vertices and edges in the graph are n and m respectively, the following code in C++ tells you how to generate the graph and store the i-th edge between the vertex $u[i]$ and $v[i]$ with weight $w[i]$ in corresponding arrays. You can use the code directly in your submissions.

```
unsigned long long k1, k2;

unsigned long long xorShift128Plus() {
unsigned long long k3 = k1, k4 = k2;
k1 = k4;
k3 ^= k3 << 23;
k2 = k3 ^ k4 ^ (k3 >> 17) ^ (k4 >> 26);
return k2 + k4;
}

int n, m, u[100001], v[100001];
unsigned long long w[100001];

void gen() {
    scanf("%d%d%llu%llu", &n, &m, &k1, &k2);
    for(int i = 1; i <= m; ++i) {
        u[i] = xorShift128Plus() % n + 1;
        v[i] = xorShift128Plus() % n + 1;
        w[i] = xorShift128Plus();
    }
}
```

Also, to decrease the size of the output, your code should output the answer modulo $(10^9 + 7)$.

If you have already learned how to handle that, start your show and omit all the rest of the statement.

To make sure everyone knows how to solve this problem, here Rikka would like to provide for you all an effective practice which can solve the problem and help you all get Accepted!

The first one you need to know is the Kirchhoff's matrix tree theorem. Given an undirected graph G with n vertices excluding all loops, its Laplacian matrix $L_{n\times n}$ is defined as $(D - A)$, where D is the degree matrix and A is the adjacency matrix of the graph. More precisely, in the matrix L the entry $l_{i,j}$ $(i \neq j)$

equals to $-m$ where m is the number of edges between the i-th vertex and the j-th vertex, and $L_{i,i}$ equals to the degree of the i-th vertex. Next, construct a matrix L^* by deleting any row and any column from L, for example, deleting row 1 and column 1. The Kirchhoff's matrix tree theorem shows that the number of spanning trees is exactly the determinant of L^*, which can be computed in polynomial time.

Now let me explain an algorithm that counts the number of MSTs. The algorithm breaks up the Kruskal's algorithm for MST into a series of blocks, each of which consists of a sequence of operations about adding edges in a same weight into a multigraph (where a multigraph is a graph, two vertices of which may be connected by more than one edge) whose vertices are components that have been built through the previous block of operations.

Precisely speaking, let's label the multigraph that has been built after the i-th block of operations as G_i. Without loss of generality, let's consider the 0-th block which has no operation and let G_0 be an empty graph with n isolated vertices. The i-th block of operations squeezes vertices in G_{i-1} connected by edges in this block into a single vertex. The result is exactly the graph G_i.

If you know the cardinal principle of Kruskal's algorithm pretty well, you may find that the number of MSTs is the product of the numbers of spanning trees in every component of the graph for each block-defining weight. Actually, the number of edges for a certain weight is fixed in all MSTs, based on the greedy-choice strategy in Kruskal's algorithm. Finally, the Kirchhoff's matrix tree theorem helps you compute the numbers of spanning trees for graphs.

Input

The input contains several test cases, and the first line contains a single integer T ($1 \le T \le 100$), the number of test cases.

For each test case, the only line contains four integers n ($1 \le n \le 10^5$), m ($m = 10^5$), k_1 and k_2 ($10^8 \le k_1, k_2 \le 10^{12}$), where k_1 and k_2 are chosen randomly except for the sample.

Output

For each test case, output a single line with a single number, the answer modulo $(10^9 + 7)$.

Example

standard input	standard output
1 2 100000 123456789 987654321	575673759

Note

Since the generator code is only provided for C++, Rikka strongly suggests you all solve the problem using C or C++ instead of other programming languages.