Tripartite Matching

HackerRank

You are given 3 unweighted, undirected graphs, G_1, G_2, and G_3, with n vertices each, where the k^{th} graph has m_k edges and the vertices in each graph are numbered from 1 through n. Find the number of ordered triples (a, b, c), where $1 \leq a, b, c \leq n$, $a \neq b, b \neq c, c \neq a$, such that there is an edge (a, b) in G_1, an edge (b, c) in G_2, and an edge (c, a) in G_3.

Input Format

The first line contains single integer, n, denoting the number of vertices in the graphs. The subsequent lines define G_1, G_2, and G_3. Each graph is defined as follows:

1. The first line contains an integer, m, describing the number of edges in the graph being defined.
2. Each line i of the m subsequent lines (where $1 \leq i \leq m$) contains 2 space-separated integers describing the respective nodes, u_i and v_i connected by edge i.

Constraints

- $n \leq 10^5$
- $m_k \leq 10^5$, and $k \in \{1, 2, 3\}$
- Each graph contains no cycles and any pair of directly connected nodes is connected by a maximum of 1 edge.

Output Format

Print a single integer denoting the number of distinct (a, b, c) triples as described in the *Problem Statement* above.

Sample Input

```
3
2
1 2
2 3
3
1 2
1 3
2 3
2
1 3
2 3
```

Sample Output

```
3
```

Explanation

There are three possible triples in our *Sample Input*:

1. $(1, 2, 3)$
2. $(2, 1, 3)$
3. $(3, 2, 1)$

Thus, we print 3 as our output.