Description

Given an **undirected** graph $G = (V, E)$ where V and E represent the set of vertices and the set of edges in the graph G respectively, each vertex is numbered from 1 to $|V|$. Contestants are required to write a program that evaluates the number of 3-tuples (u, v, x) such that graph G exists a simple path (a path that passes each vertex at most once) from u to v through x, where $u < v$ and $x \neq u, v$.

For example, the answer of a chain graph with 4 vertices is 4.

Constraints

Time limit: 1.5 s / Memory limit: 512 MB

$3 \leqslant |V| \leqslant 3 \times 10^5$, $|E| \leqslant 5 \times 10^5$.

Solution

Category: Graph Theory, Dynamic Programming

It is important to observe that for any *biconnected* (2-vertex-connected) graph G, any 3-tuple (u, v, x) is legal. Supposing not, there must be at least one cut vertex on the path from u to x and v to x. To generalize the conclusion, we shrink all the biconnected components into single vertices (marked as squares) which connect to each vertex in the component (marked as circles) with an edge so that graph G will become a tree T. Then a 3--tuple (u, v, x) is legal if and only if:

1. u, v and x are in the same connected component.
2. x is on the unique path from u to v in T.
3. Otherwise, S denotes the set of vertices of a biconnected component associated with x, and u' and v' denote the first vertex in S on the path to x respectively. Therefore, $u' \neq v'$ for at least one possible biconnected component, since x may be shared by more than two biconnected components.

The conditions directly give the method to judge a 3-tuple. To count it, we need to consider two cases (assuming T is a rooted forest):

1. Directly pass the vertex x: compute an array $g[x]$ that represents the number of pairs (u, v) where both u and v are in the subtree of x and the path from u to v passes x. Note that square vertices are just auxiliary vertices and thus u and v should not be square vertices. In this case, the contribution is the sum of $g[x]$ where x is a circle vertex.
2. Considering a specific biconnected component, u, v come from different u' and v'. After considered case 1, it's easy to show that the contribution of this case is $(m-2)g[x]$, where m represents the number of vertices in the component, since $g[x]$ represents the number of pairs (u, v) and $m-2$ is the number of x that can be selected in the component.

For convenience, the attached standard program uses the technique of "reroot" so that we don't care about the case that some circle vertex is the father of a square vertex. Details can be seen in the source code.

Testdatas

Any (randomized) graphs would be fine since there seems to be few tricks to solve the problem unexpectedly, so I didn't attach any testdata in the archive.

Degenerated graphs like **chain graphs**, **trees** and **cactus graphs** are supposed to be subtasks of the problem. Methods for these graphs are relatively easy compared to the standard solution. Moreover, **disconnected** graphs with **loops** and **multiple edges** should be included to test contestants' implementations of Tarjan's algorithm.

Notes

The attached program get inputs of graph via the format to be showed below:

```
[n] [m]
[u] [v]
... (m edges (u, v))
```

where $n = |V|$ and $m = |E|$. The program outputs the answer of the given graph.

Personal Information

Name: Zhenliang Xue

Nationality: China

Affiliation: The High School Attached to Hunan Normal University, pre-university student of Fudan University

Experiences: Chinese NOI 2017 silver medalist, once participated in APIO 2017

Email: riteme@qq.com or 1412803389@qq.com