

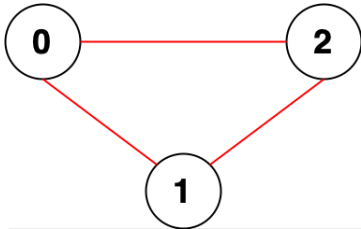
6106. Count Unreachable Pairs of Nodes in an Undirected Graph

[My Submissions \(/contest/biweekly-contest-81/problems/count-unreachable-pairs-of-nodes-in-an-undirected-graph/submissions/\)](#)
[Back to Contest \(/contest/biweekly-contest-81/\)](#)

You are given an integer n . There is an **undirected** graph with n nodes, numbered from 0 to $n - 1$. You are given a 2D integer array `edges` where `edges[i] = [ai, bi]` denotes that there exists an **undirected** edge connecting nodes a_i and b_i .

Return the **number of pairs** of different nodes that are **unreachable** from each other.

Example 1:

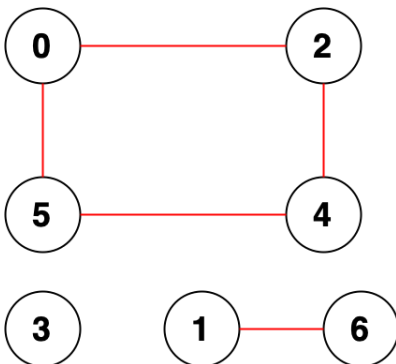


Input: $n = 3$, `edges = [[0,1],[0,2],[1,2]]`

Output: 0

Explanation: There are no pairs of nodes that are unreachable from each other. Therefore, we return 0.

Example 2:



Input: $n = 7$, `edges = [[0,2],[0,5],[2,4],[1,6],[5,4]]`

Output: 14

Explanation: There are 14 pairs of nodes that are unreachable from each other: `[[0,1],[0,3],[0,6],[1,2],[1,3],[1,4],[1,5],[2,3],[2,6],[3,4],[3,5],[3,6],[4,6],[5,6]]`. Therefore, we return 14.

Constraints:

- $1 \leq n \leq 10^5$
- $0 \leq \text{edges.length} \leq 2 \times 10^5$
- `edges[i].length == 2`
- $0 \leq a_i, b_i < n$
- $a_i \neq b_i$
- There are no repeated edges.

JavaScript



```

1 function DJSet(n) {
2   // parent[i] < 0, -parent[i] is the group size which root is i. example: (i -> parent[i] -> parent[parent[i]] ->
   parent[parent[parent[i]]] ...)
3   // parent[i] >= 0, i is not the root and parent[i] is i's parent. example: (... parent[parent[parent[i]]] ->
   parent[parent[i]] -> parent[i] -> i)
4   let parent = Array(n).fill(-1);

```

```


5   return { find, union, count, equiv, par }
6   function find(x) {
7       return parent[x] < 0 ? x : parent[x] = find(parent[x]);
8   }
9   function union(x, y) {
10      x = find(x);
11      y = find(y);
12      if (x == y) return false;
13      if (parent[x] < parent[y]) [x, y] = [y, x];
14      parent[x] += parent[y];
15      parent[y] = x;
16      return true;
17  }
18  function count() { // total groups
19      return parent.filter(v => v < 0).length;
20  }
21  function equiv(x, y) { // isConnected
22      return find(x) == find(y);
23  }
24  function par() {
25      return parent;
26  }
27 }
28
29 const countPairs = (n, edges) => {
30     let ds = new DJSet(n);
31     for (const [x, y] of edges) ds.union(x, y);
32     let d = [];
33     for (const x of ds.par()) {
34         if (x < 0) d.push(-x);
35     }
36     let len = d.length, res = 0;
37     for (let i = 0; i < len; i++) {
38         for (let j = i + 1; j < len; j++) {
39             res += d[i] * d[j];
40         }
41     }
42     return res;
43 };

```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)Submission Result: **Accepted** (/submissions/detail/730943039/) ⓘ[More Details > \(/submissions/detail/730943039/\)](#)

Share your acceptance!

Copyright © 2022 LeetCode

[Help Center \(/support\)](#) | [Jobs \(/jobs\)](#) | [Bug Bounty \(/bugbounty\)](#) | [Online Interview \(/interview/\)](#) | [Students \(/student\)](#) | [Terms \(/terms\)](#) | [Privacy Policy \(/privacy\)](#) [United States \(/region\)](#)