Started on	Sunday, 24 March 2024, 9:58 AM
State	Finished
Completed on	Sunday, 24 March 2024, 10:42 AM
Time taken	43 mins 40 secs
Grade	**10.00** out of 10.00 (**100**%)

Question 1

Correct

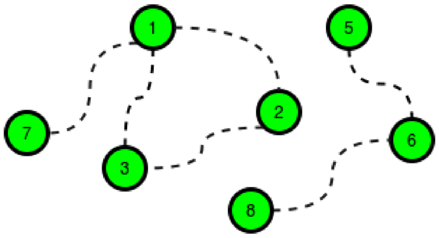
Mark 10.00 out of 10.00

Determine the minimum cost to provide library access to all citizens of HackerLand. There are n cities numbered from 1 to n. Currently there are no libraries and the cities are not connected. Bidirectional roads may be built between any city pair listed in $cities$. A citizen has access to a library if:

- Their city contains a library.
- They can travel by road from their city to a city containing a library.

Example

The following figure is a sample map of HackerLand where the dotted lines denote possible roads:



$c\_road = 2$
$c\_lib = 3$
$cities = [[1,7],[1,3],[1,2],[2,3],[5,6],[6,8]]$

The cost of building any road is $cc\_road = 2$, and the cost to build a library in any city is $c\_lib = 3$. Build 5 roads at a cost of $5 \times 2 = 10$ and 2 libraries for a cost of 6. One of the available roads in the cycle $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ is not necessary.

There are q queries, where each query consists of a map of HackerLand and value of c_lib and c_road. For each query, find the minimum cost to make libraries accessible to all the citizens.

Function Description

Complete the function *roadsAndLibraries* in the editor below.
roadsAndLibraries has the following parameters:

- *int n*: integer, the number of cities
- *int c_lib*: integer, the cost to build a library
- *int c_road*: integer, the cost to repair a road
- *int cities[m][2]*: each $cities[i]$ contains two integers that represent cities that can be connected by a new road

Returns

- *int*: the minimal cost

Input Format

The first line contains a single integer q, that denotes the number of queries.

The subsequent lines describe each query in the following format:
- The first line contains four space-separated integers that describe the respective values of n, m, c_lib and c_road, the number of cities, number of roads, cost of a library and cost of a road.
- Each of the next m lines contains two space-separated integers, $u[i]$ and $v[i]$, that describe a bidirectional road that can be built to connect cities $u[i]$ and $v[i]$.

Constraints

- $1 \leq q \leq 10$
- $1 \leq n \leq 10^5$
- $0 \leq m \leq min(10^5, \frac{n \cdot (n-1)}{2})$
- $1 \leq c\_road, c\_lib \leq 10^5$
- $1 \leq u[i], v[i] \leq n$
- Each road connects two distinct cities.

For example:

Input	Result

2
3 3 2 1
1 2
3 1
2 3
6 6 2 5
1 3
3 4
2 4
1 2
2 3
5 6
	4
	12

Answer: (penalty regime: 0 %)

Reset answer

```
#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);

long roadsAndLibraries(int n, int c_li
    if (c_lib <= c_road) {
        return static_cast<long>(n) *
    }

    vector<bool> visited(n + 1, false
    vector<vector<int>> adj_list(n + 1

    for (auto& road : cities) {
        adj_list[road[0]].push_back(ro
        adj_list[road[1]].push_back(ro
    }

    long total_cost = 0;

    for (int i = 1; i <= n; ++i) {
        if (!visited[i]) {
            long num_cities = 0;
            long num_roads = 0;
            queue<int> q;
            q.push(i);
            visited[i] = true;
            visited[i] = true;
            while (!q.empty()) {
                int curr = q.front();
                q.pop();
                ++num_cities;
                for (int neighbor : ac
                    if (!visited[neigh
                        visited[neighb
                        q.push(neighbo
                        ++num_roads;
                    }
                }
            }
            total_cost += c_lib + c_ro
        }
    }

    return total_cost;
}

int main() {
    string q_temp;
    getline(cin, q_temp);
```

	Input	Expected	Got	
✓	2 3 3 2 1 1 2 3 1 2 3 6 6 2 5 1 3 3 4 2 4 1 2 2 3 5 6	4 12	4 12	✓
✓	5 9 2 91 84 8 2 2 9 5 9 92 23 2 1 5 3 5 1 3 4 3 1 5 4 4 1 5 2 4 2 8 3 10 55 6 4 3 2 7 1 1 0 5 3 2 0 102 1	805 184 80 5 204	805 184 80 5 204	✓
✓	1 5 3 6 1 1 2 1 3 1 4	15	15	✓

Passed all tests! ✓

▸ **Show/hide question author's solution (Cpp)**

Correct

Marks for this submission: 10.00/10.00.