

BurningBridges.cpp

```

//
// Created by juan on 1/05/17.
//

#include<iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <map>

using namespace std;

vector<vector<pair<int, int> > > graph;
vector<int> answer, disc, parent, low;
vector<bool> visited;

//These are for machete
set<int> nodes;
set<int> banned;
map<int, int> m;

int N, M, t;

void dfs(int u) {
    visited[u] = 1;
    disc[u] = low[u] = ++t;
    int neighbor, edge;
    for (int i = 0; i < graph[u].size(); i++){
        neighbor = graph[u][i].first;
        edge = graph[u][i].second;
        if ( visited[ neighbor ] == 0 ){
            parent[neighbor] = u;
            dfs(neighbor);
            low[u] = min(low[u], low[neighbor]);
            if (low[neighbor] > disc[u])
                if (banned.count(edge) != 1)
                    answer.push_back(edge);
        }
        else if (neighbor != parent[u])
            low[u] = min(low[u], disc[neighbor]);
    }
}

int main(){
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    int T, a, b, tmp;
    cin >> T;
    for (int c = 1; c <= T; c++){
        cin >> N >> M;
        graph.assign(N+1, vector<pair<int, int> >());
        low.assign(N+1, 0);
        disc.assign(N+1, 0);
        parent.assign(N+1, 0);
        visited.assign(N+1, 0);
        answer.clear();
        m.clear();
        banned.clear();
        nodes.clear();
        for (int i = 1; i <= M; i++){

```

```
    cin >> a >> b;
    graph[a].push_back(make_pair(b, i));
    graph[b].push_back(make_pair(a, i));

    if (a > b)
        tmp = a + b*16384;
    else
        tmp = b + a*16384;
    if (nodes.count(tmp) == 0) {
        nodes.insert(tmp);
        m[tmp] = i;
    }
    else {
        banned.insert(i);
        banned.insert(m[tmp]);
    }
}
t = 0;
dfs(1);
sort(answer.begin(), answer.end());
cout << answer.size();
if (answer.size() > 0) {
    cout << endl;
    for (int i = 0; i < answer.size() - 1; i++)
        cout << answer[i] << " ";
    cout << answer[answer.size() - 1];
}
if (c != T)
    cout << "\n\n";
}
```