

Highways

Swiss Olympiad in Informatics

February 12, 2018

Task description

N cities are connected by M roads. You want to upgrade some roads to highways so that every city is incident to an **odd** number of highways.

- Can this be done?
- Which roads should be upgraded?

Handshaking lemma

The number of nodes of odd degree is even.

- Proof: The sum of degrees is twice the number of edges.

Dfs solution

Connected components can be solved separately.
Dfs gives us a dfs tree. We'll only use tree-edges.

Dfs solution

Connected components can be solved separately.

Dfs gives us a dfs tree. We'll only use tree-edges.

- For a non-root node, we can use the parent edge to make its degree odd.
- The only problematic node is the root.

Root problems (single component)

If the root has got even degree.

- All other nodes have got odd degree.

⇒ There's an even number of non-root nodes, so the total number of nodes is odd.

In that case there's no solution (Handshaking lemma).

Code: dfs

```
1  vector<vector<int>> g;
2  vector<pair<int,int>> edges;
3  vector<bool> vis;
4  int dfs(int x,int p){
5      if(vis[x]) return 0;
6      vis[x]=true;
7      int r=0;
8      for(int y:g[x]) r+=dfs(y,x);
9      if(!(r%2)){
10         edges.emplace_back(p,x);
11         return 1;
12     }
13     return 0;
14 }
```

Code: main

```
1 ios_base::sync_with_stdio(0),cin.tie(0);
2 int n,m; cin>>n>>m;
3 vis.resize(n); g.resize(n);
4 for(int i=0;i<m;++i){
5     int a,b; cin>>a>>b;
6     g[a].push_back(b);
7     g[b].push_back(a);
8 }
9 for(int i=0;i<n;++i)
10     if(!vis[i] && dfs(i,-1)) {
11         cout<<"NONE\n";
12         return 0;
13     }
14 cout<< edges.size() <<'\n';
15 for(auto &edge:edges)
16     cout<<edge.first<<" " <<edge.second<<'\n';
```