

```

class Solution {
public:
    void dfs(int i, int p, vector<int>& v, int v1[], int v2[], int& cnt,
vector<int>adj[], vector<vector<int>>& ans){
        v[i]=1;
        v1[i]=cnt;
        v2[i]=cnt;
        cnt++;
        for(auto it: adj[i]){
            if(it==p){continue;}
            if(v[it]){v1[i]=min(v1[i], v1[it]);}
            else{
                dfs(it, i, v, v1, v2, cnt, adj, ans);
                v1[i]=min(v1[i], v1[it]);
                if(v1[it]>v2[i]){
                    ans.push_back({i, it});
                }
            }
        }
    }
    vector<vector<int>> criticalConnections(int n, vector<vector<int>>&
connections) {
        vector<int>adj[n];
        for(int i=0; i<connections.size(); i++){
            adj[connections[i][0]].push_back(connections[i][1]);
            adj[connections[i][1]].push_back(connections[i][0]);
        }
        int cnt=1;
        vector<int>v(n, 0);
        int v1[n], v2[n];
        vector<vector<int>>ans;
        dfs(0, -1, v, v1, v2, cnt, adj, ans);
        return ans;
    }
};

```