Team Notebook

June 23, 2020

Contents

1 DFS

1.1 DFS on graph

```
#include <bits/stdc++.h>
using namespace std;
vector<int> adj[10001];
bool vis[10001]={0};
int ii;
void dfs(int v)
{  vis[v]=true; ii++;
   cout<<v;
   for(auto u: adj[v])
   if(!vis[u])
   dfs(u);
}
int main() {
 int n,m,u,v;
 cin>>n>>m;
 for(int i=0;i<m;i++)
 { cin>>u>>v;
   adj[u].push_back(v);
   adj[v].push_back(u);
 }
 dfs(1);
 return 0;
}
```

1.2 DFS on tree

```
#include <bits/stdc++.h>
using namespace std;
vector<int> adj[10001];
int ii;
void dfs(int v, int par){
 cout<<v<<" ";
 for(auto u: adj[v])
  {  if (u == par) continue;
   dfs(u, v);
  }
}
int main() {
 int n,u,v;
 cin>>n;
 for(int i=0;i<n-1;i++)
 { cin>>u>>v;
   adj[u].push_back(v);
   adj[v].push_back(u);
 }
 dfs(1,-1);
 return 0;
}
```