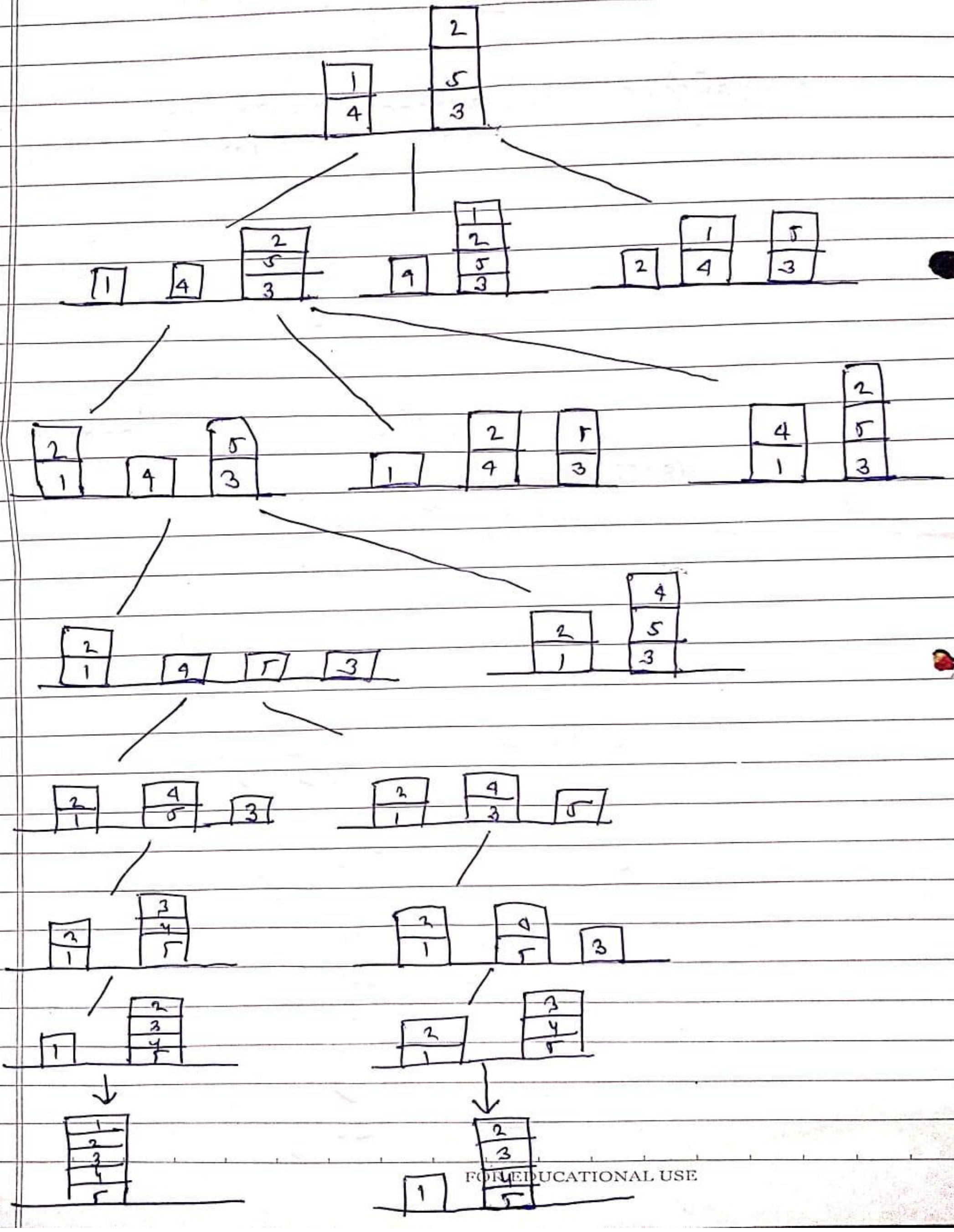
Name : Vishal Kanhaiya Jha

AI/ML Assignment 1

01/Sep/2022
Thursday

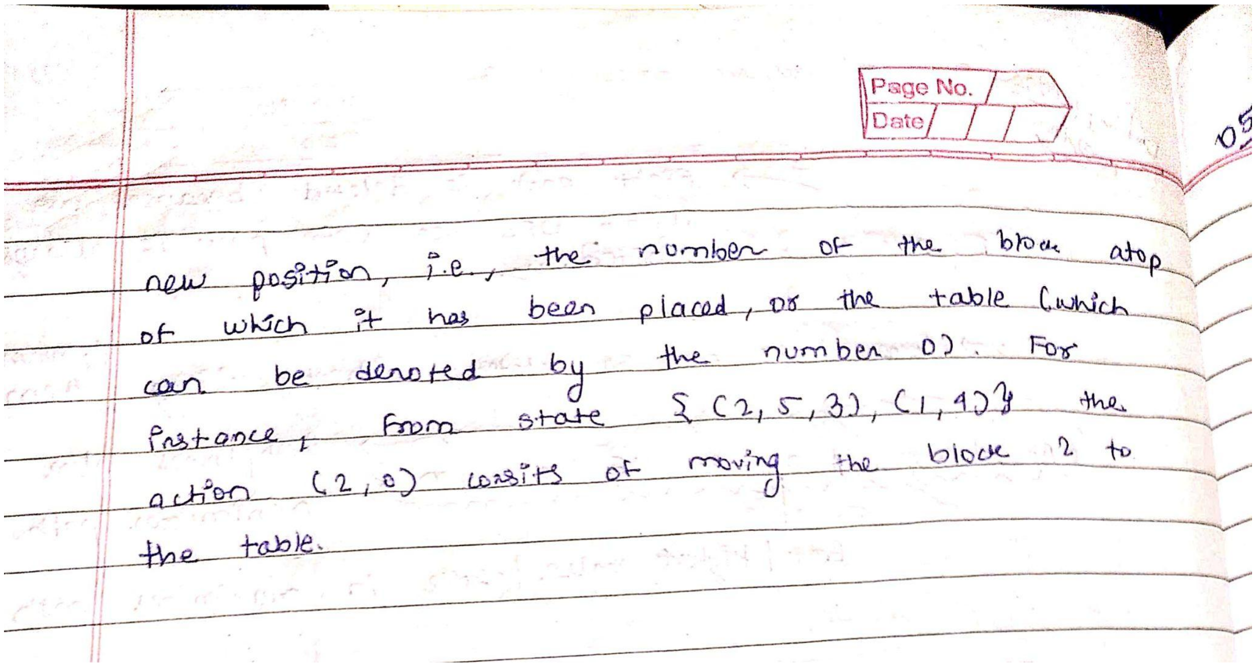
Q. 1



Q d) The state space is made up of the set of distinct arrangements of the five blocks into one or more piles. A state can be represented by a set of piles, and each pile can be represented as a sequence of block numbers, where the numbers from left to right correspond, e.g. to block from the top to the bottom of the pile. For instance, the two equivalent configuration of piles shown on the left and in the middle of the above figure are represented by the set $\{(2,5,3), (1,4)\}$ Accordingly, the goal state is represented by $(1,2,3,4,5)$.

The actions can be formally described as follows : given a state $\{(b_{1,1}, \ldots\ldots, b_{1,n_1}), \ldots\ldots, (b_{p,1}, \ldots\ldots, b_{p,n_p})\}$, where p denotes the number of piles $(1 \leq p \leq 5)$ & n_k the number of blocks in the kth pile ($n_k \geq 1$ for each k, and $\sum_{k=1}^{p} n_k = 5$), the actions consist of moving one of the p blocks $b_{k,1}$ (on the top of one of the piles) either to the table, thus generating a new pile (only if the original pile contains more than one block, i.e., $n_k > 1$), or atop one of the other $p-1$ piles (if any).

Actions can be implemented as a successor function SF : it receives as an argument the description s of a state, and returns all the pairs (s', a) where s' is one of the states obtained as described above, and a the description of the corresponding action. Each action can be described by indicating the number of the block that has been moved, $b_{k,1}$, and its



Page No.
Date

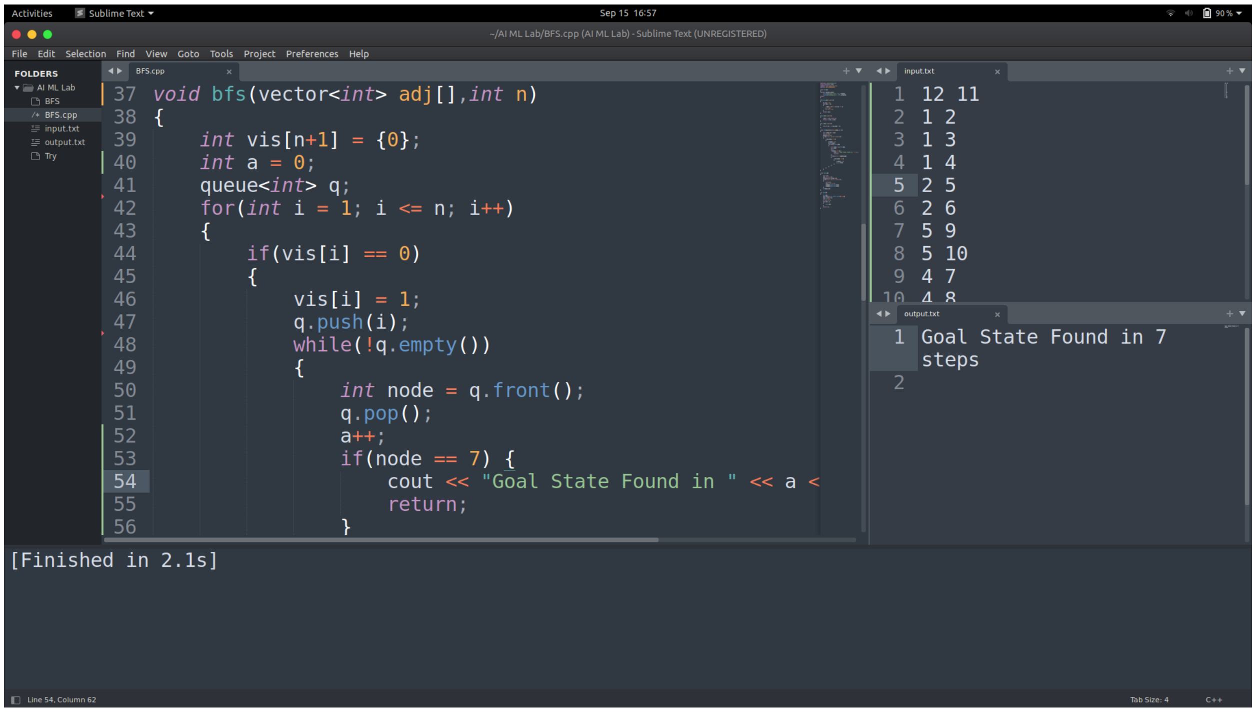
new position, i.e., the number of the block atop of which it has been placed, or the table (which can be denoted by the number 0). For instance, from state $\{(2,5,3),(1,4)\}$ the action $(2,0)$ consists of moving the block 2 to the table.

Q.2)

Q.2

$$\begin{tikzcd}[column sep=small]
 & \begin{array}{c}\text{ABABAECCEC}\ (0+10)\end{array} \arrow[dl, "AB=BC"'] \arrow[d] \arrow[dr] \arrow[drr] & & \\
\begin{array}{c}\text{BCABAECCEC}\\(1+10)\end{array} & \begin{array}{c}\text{ABBCAECCEC}\\(1+10)\end{array} & \begin{array}{c}\text{ABABACCEC}\\(1+9)\end{array} \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] & \begin{array}{c}\text{ABABAECCC}\\(1+9)\end{array} \\
\begin{array}{c}\text{BCABACCEC}\\(2+9)\end{array} & \begin{array}{c}\text{ABBCACCEC}\\(2+9)\end{array} & \begin{array}{c}\text{ABABECEC}\\(2+8)\end{array} \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] & \begin{array}{c}\text{ABABACCC}\\(2+8)\end{array} \\
\begin{array}{c}\text{BCABECEC}\\(3+8)\end{array} & \begin{array}{c}\text{ABBCECEC}\\(3+8)\end{array} & \begin{array}{c}\text{ABABCEC}\\(3+7)\end{array} \arrow[dll] \arrow[dl] \arrow[d] & \begin{array}{c}\text{ABABECC}\\(3+7)\end{array} \\
\begin{array}{c}\text{BCABCEC}\\(4+7)\end{array} & \begin{array}{c}\text{ABBCCEC}\\(4+7)\end{array} & \begin{array}{c}\text{ABABCC}\\(4+6)\end{array} \arrow[dl] \arrow[d] & \\
 & \begin{array}{c}\text{BCABCC}\\(5+6)\end{array} \arrow[d] & \begin{array}{c}\text{ABBCCC}\\(5+6)\end{array} \arrow[d] \arrow[dr] & \\
 & \underset{\times}{\text{BCBCCC}} & \text{AECCC} \arrow[d] & \text{BCBCCC} \\
 & & \text{ACCC} \arrow[d] & \\
 & & \text{ECC} \arrow[d] & \\
 & & \text{CC} &
\end{tikzcd}$$

Q.3)
BFS :
```
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
#define mod 1000000007

void file()
{
#ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
}

ll binpow(ll a,ll b)
{
    ll ans = 1;
    while(b > 0)
    {
        if((b & 1) == 1) ans *= a;
        a *= a;
        b = b >> 1;
    }
    return ans;
}

ll gcd(ll a,ll b)
{
        if(b == 0) return a;
        return gcd(b, a%b);
}

ll lcm(ll a,ll b)
{
        return (a / gcd(a,b)) * b;
}

void bfs(vector<int> adj[],int n)
{
        int vis[n+1] = {0};
        int a = 0;
        queue<int> q;
        for(int i = 1; i <= n; i++)
        {
                if(vis[i] == 0)
                {
                        vis[i] = 1;
                        q.push(i);
                        while(!q.empty())
                        {
                                int node = q.front();
                                q.pop();
                                a++;
                                if(node == 7) {
                                        cout << "Goal State Found in " << a << " steps" << endl;
                                        return;
                                }
                                for(auto i : adj[node])
                                {
                                        if(vis[i] == 0)
                                        {
                                                vis[i] = 1;
                                                q.push(i);
                                        }
                                }
                        }
                }
        }
}

void solve()
{
        int n,m;
        cin >> n >> m;
        vector<int> adj[n+1];
        for(int i = 1; i <= m; i++)
        {
                int u,v;
                cin >> u >> v;
                adj[u].push_back(v);
                adj[v].push_back(u);
        }
        bfs(adj,n);
}

int main()
{
        file();
        ios_base::sync_with_stdio(false);
    cin.tie(NULL);
        int t = 1;
        // cin >> t;
        while(t--)
        {
                solve();
        }
        return 0;
}
```



```
void bfs(vector<int> adj[],int n)
{
    int vis[n+1] = {0};
    int a = 0;
    queue<int> q;
    for(int i = 1; i <= n; i++)
    {
        if(vis[i] == 0)
        {
            vis[i] = 1;
            q.push(i);
            while(!q.empty())
            {
                int node = q.front();
                q.pop();
                a++;
                if(node == 7) {
                    cout << "Goal State Found in " << a <
                    return;
                }
```

```
12 11
1 2
1 3
1 4
2 5
2 6
5 9
5 10
4 7
4 8
```

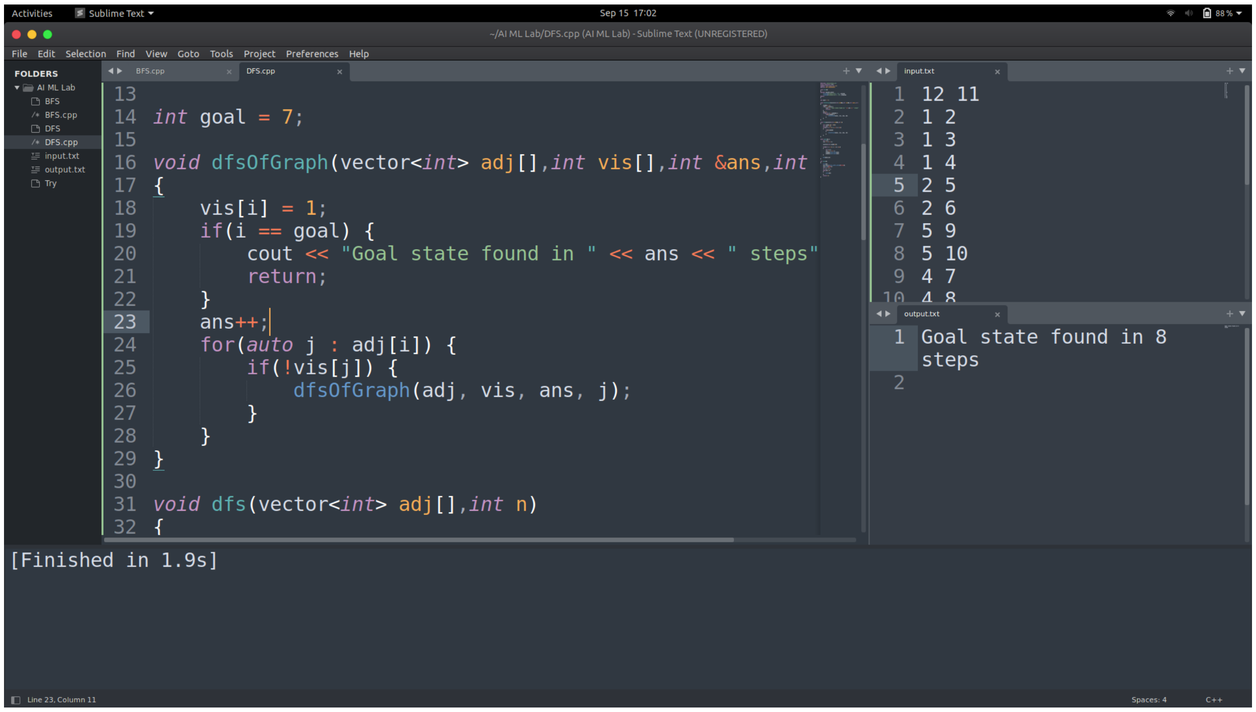
```
Goal State Found in 7
steps
```

```
[Finished in 2.1s]
```

DFS :

```
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
#define mod 1000000007

void file()
{
#ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
}
```

```
int goal = 7;

void dfsOfGraph(vector<int> adj[],int vis[],int &ans,int i)
{
    vis[i] = 1;
    if(i == goal) {
        cout << "Goal state found in " << ans << " steps" << endl;
        return;
    }
    ans++;
    for(auto j : adj[i]) {
        if(!vis[j]) {
            dfsOfGraph(adj, vis, ans, j);
        }
    }
}

void dfs(vector<int> adj[],int n)
{
    int vis[n+1] = {0};
    int ans = 0;
    for(int i = 1; i <= n; i++)
    {
        if(!vis[i])
        {
            dfsOfGraph(adj, vis, ans, i);
        }
    }
}

void solve() {
    int n,m;
    cin >> n >> m;

    vector<int> adj[n+1];

    for(int i = 0; i < m; i++)
    {
        int u,v;
        cin >> u >> v;
        adj[u].push_back(v);
        adj[v].push_back(u);
    }
    dfs(adj,n);
}

int main()
{
    file();
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int t = 1;
    // cin >> t;
    while(t--)
    {
        solve();
    }
    return 0;
}
```



```

int goal = 7;

void dfsOfGraph(vector<int> adj[],int vis[],int &ans,int
{
    vis[i] = 1;
    if(i == goal) {
        cout << "Goal state found in " << ans << " steps"
        return;
    }
    ans++;
    for(auto j : adj[i]) {
        if(!vis[j]) {
            dfsOfGraph(adj, vis, ans, j);
        }
    }
}

void dfs(vector<int> adj[],int n)
{
```

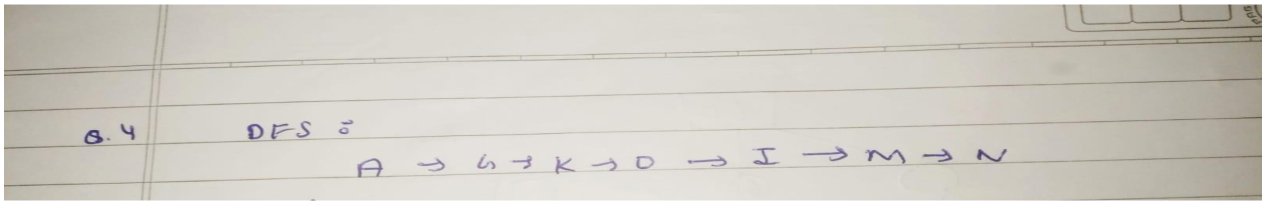
```
12 11
1 2
1 3
1 4
2 5
2 6
5 9
5 10
4 7
4 8
```

```
Goal state found in 8
steps

```

```
[Finished in 1.9s]
```

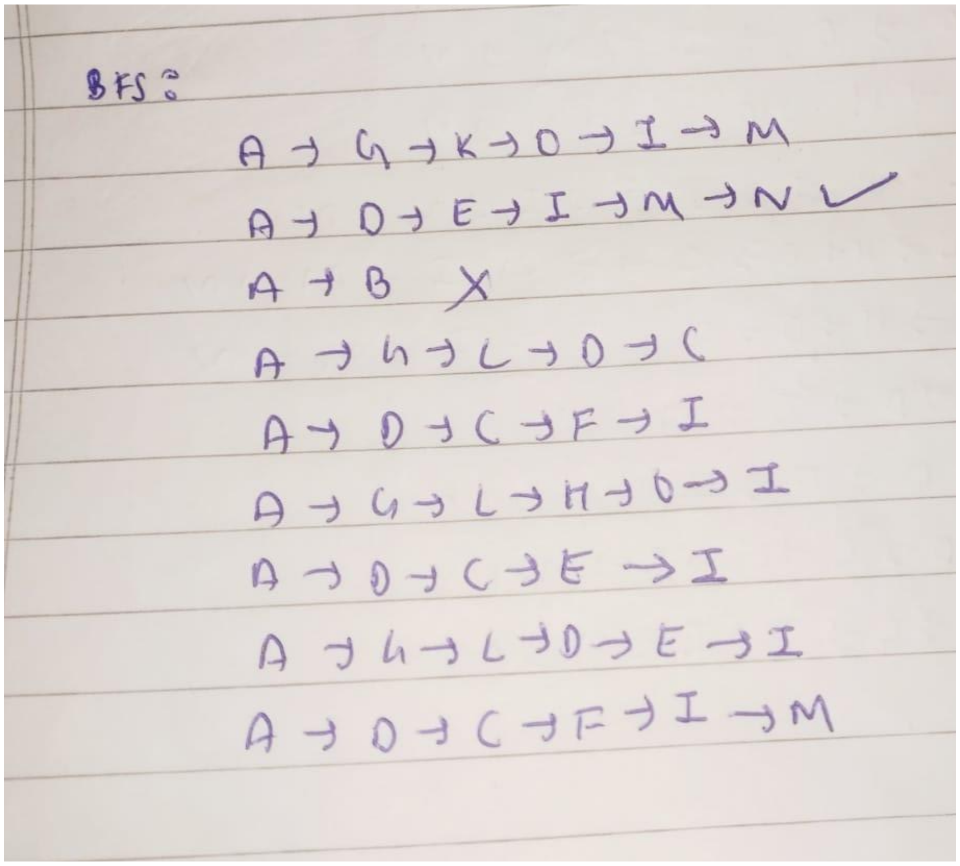
From The result, we can see that BFS took less time as compared to DFS method, so we can say that BFS is better in this case as it reduces time complexity !

Q.4)



Q.4 DFS :

A → G → K → D → I → M → N



BFS :

A → G → K → D → I → M

A → D → E → I → M → N ✓

A → B ✗

A → G → L → D → C

A → D → C → F → I

A → G → L → H → D → I

A → D → C → E → I

A → G → L → D → E → I

A → D → C → F → I → M

Q.5)

BFS :

```
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
#define mod 1000000007

void file()
{
#ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
}
```

```
ll binpow(ll a,ll b)
{
    ll ans = 1;
    while(b > 0)
    {
        if((b & 1) == 1) ans *= a;
        a *= a;
        b = b >> 1;
    }
    return ans;
}

ll gcd(ll a,ll b)
{
        if(b == 0) return a;
        return gcd(b, a%b);
}

ll lcm(ll a,ll b)
{
        return (a / gcd(a,b)) * b;
}

vector<int> bfs(vector<int> adj[],int n)
{
        int vis[n+1] = {0};
        queue<int> q;
        vector<int> ans;
        for(int i = 1; i <= n; i++)
        {
                if(vis[i] == 0)
                {
                        vis[i] = 1;
                        q.push(i);
                        ans.push_back(i);
                        while(!q.empty())
                        {
                                int node = q.front();
                                q.pop();

                                for(auto i : adj[node])
                                {
                                        if(vis[i] == 0)
```

```
                                                {
                                                        vis[i] = 1;
                                                        q.push(i);
                                                        ans.push_back(i);
                                                        if(i == 7) return ans;
                                                }
                                        }
                                }
                        }
                }
                return ans;
        }

        void solve()
        {
                int n,m;
                cin >> n >> m;
                vector<int> adj[n+1];
                for(int i = 1; i <= n; i++)
                {
                        int u,v;
                        cin >> u >> v;
                        adj[u].push_back(v);
                        adj[v].push_back(u);
                }
                vector<int> ans = bfs(adj,n);
                for(auto i : ans) cout << i << " ";
                cout << endl;
        }

        int main()
        {
                file();
                ios_base::sync_with_stdio(false);
           cin.tie(NULL);
                int t = 1;
                // cin >> t;
                while(t--)
                {
                        solve();
                }
                return 0;
        }
```

DFS :

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
#define mod 1000000007

void file()
{
#ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
}

bool flag = true;
void dfsOfGraph(vector<int> adj[],int vis[],vector<int>& ans,int i)
{
    if(flag == false) return;
    ans.push_back(i);
    vis[i] = 1;
    if(i == 7) {
        flag = false;
        return;
    }
    for(auto j : adj[i])
    {
        if(!vis[j])
        {
            dfsOfGraph(adj,vis,ans,j);
        }
    }
}

vector<int> dfs(vector<int> adj[],int n)
{
    int vis[n+1] = {0};
    vector<int> ans;
    for(int i = 1; i <= n; i++)
    {
        if(!vis[i])
        {
            dfsOfGraph(adj,vis,ans,i);
        }
    }
    return ans;
}

void solve() {
    int n,m;
    cin >> n >> m;

    vector<int> adj[n+1];

    for(int i = 0; i < m; i++)
    {
        int u,v;
        cin >> u >> v;
        adj[u].push_back(v);
        adj[v].push_back(u);
    }

    vector<int> ans = dfs(adj,n);
    for(auto i : ans) cout << i << " ";
    cout << endl;
}

int main()
{
    file();
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int t = 1;
    // cin >> t;
    while(t--)
    {
        solve();
    }
    return 0;
}
```