AI/ML

2020BTECS00087

OMKAR OGALE

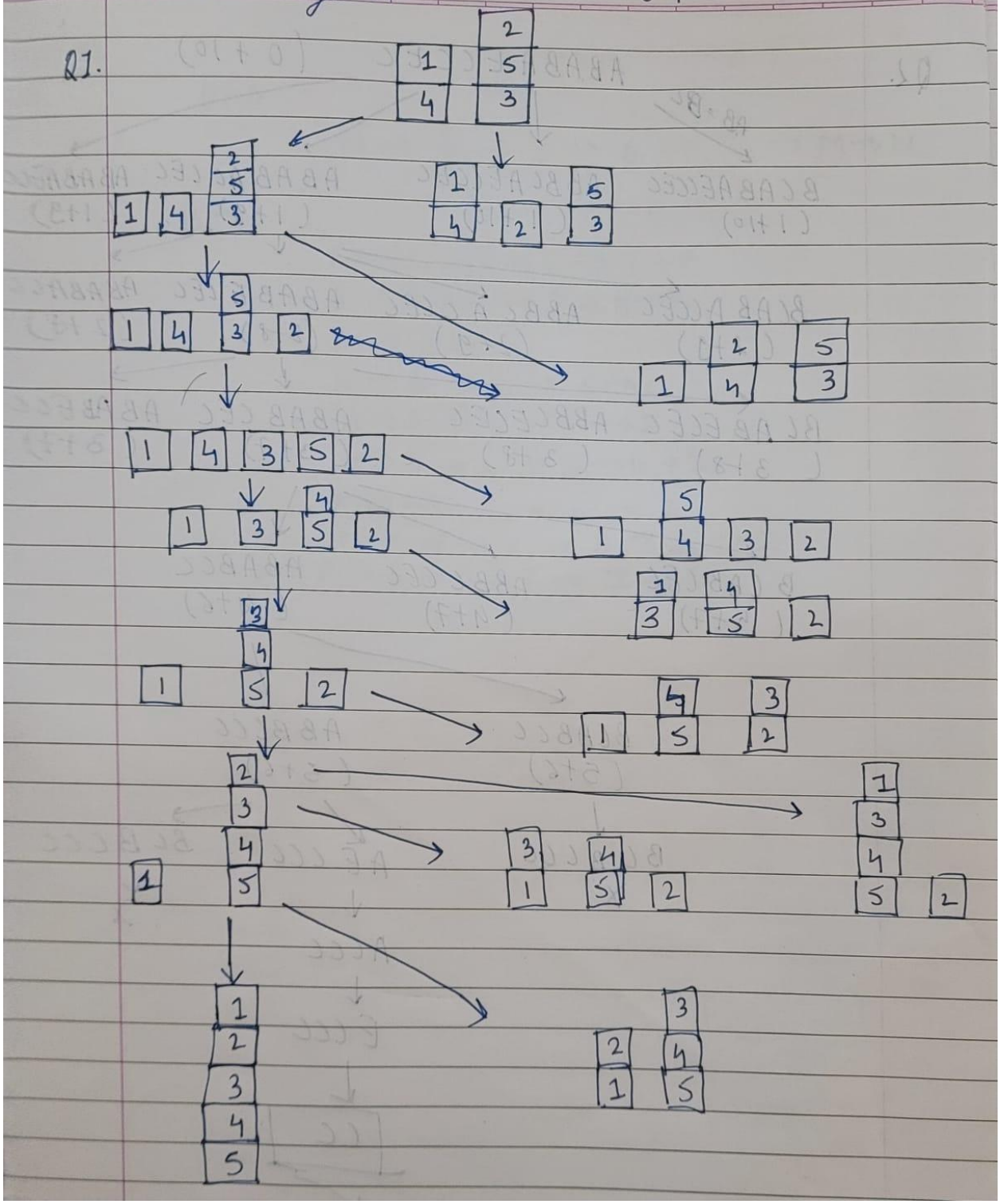
Assignment 1

Page No.:

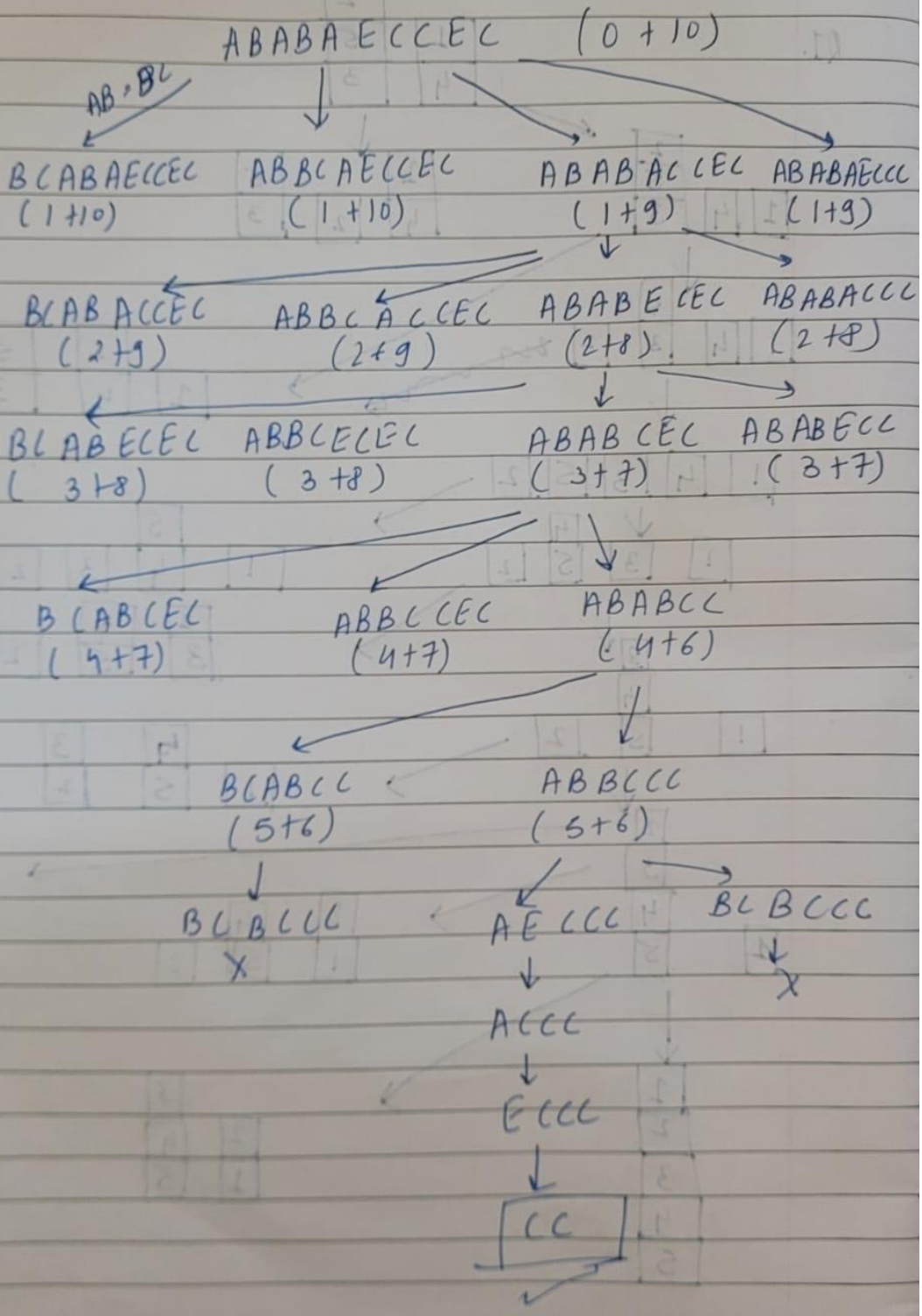
Date:

YOUVA

Q1.



Q2.


ABABAECCEC (0+10)
AB, BC
BCABAECCEC (1+10)
ABBCAECCEC (1+10)
ABABACCEC (1+9)
ABABAECCC (1+9)
BCABACCEC (2+9)
ABBCACCEC (2+9)
ABABECEC (2+8)
ABABACCC (2+8)
BCABECEC (3+8)
ABBCECEC (3+8)
ABABCEC (3+7)
ABABECC (3+7)
BCABCEC (4+7)
ABBCCEC (4+7)
ABABCC (4+6)
BCABCC (5+6)
ABBCCC (5+6)
BCBCCC
X
AECCC
ACCC
ECCC
CC

Q3)
A - BFS

```cpp
/* Editor: Omkar Ugale
DATE - 17-Aug-2022 TIME - 07:48:42*/
#include <bits/stdc++.h>

using namespace std;
const int N = 1e4 + 1;
int vis[N] = {0};
// int level[N] = {0};
int bbfs[N];
void bfs(int node, vector<int> v[])
{
    queue<int> q;
    q.push(node);
    vis[node] = 1;
    // level[node]++;
    int i = 0;
    while (!q.empty())
    {

        int t = q.front();
        bbfs[i++] = t;
        q.pop();
        for (auto it : v[t])
        {
            if (!vis[it])
            {
                vis[it] = 1;
                q.push(it);
                // level[it] = level[t] + 1;
            }
        }
    }
}
int main()
{
    int n, m;
    cin >> n >> m;
```

```
    vector<int> v[n + 1];
    for (int i = 0; i < m; i++)
    {
        int a, b;
        cin >> a >> b;
        v[a].push_back(b);
        v[b].push_back(a);
    }
    bfs(1, v);
    for (int i = 0; i <= n; i++)
        cout << bbfs[i] << " ";
    cout << endl;
    // for (int i = 1; i <= n; i++)
    //     cout <<i<<" "<< level[i] << ", ";
    // cout << endl;

    return 0;
}
```

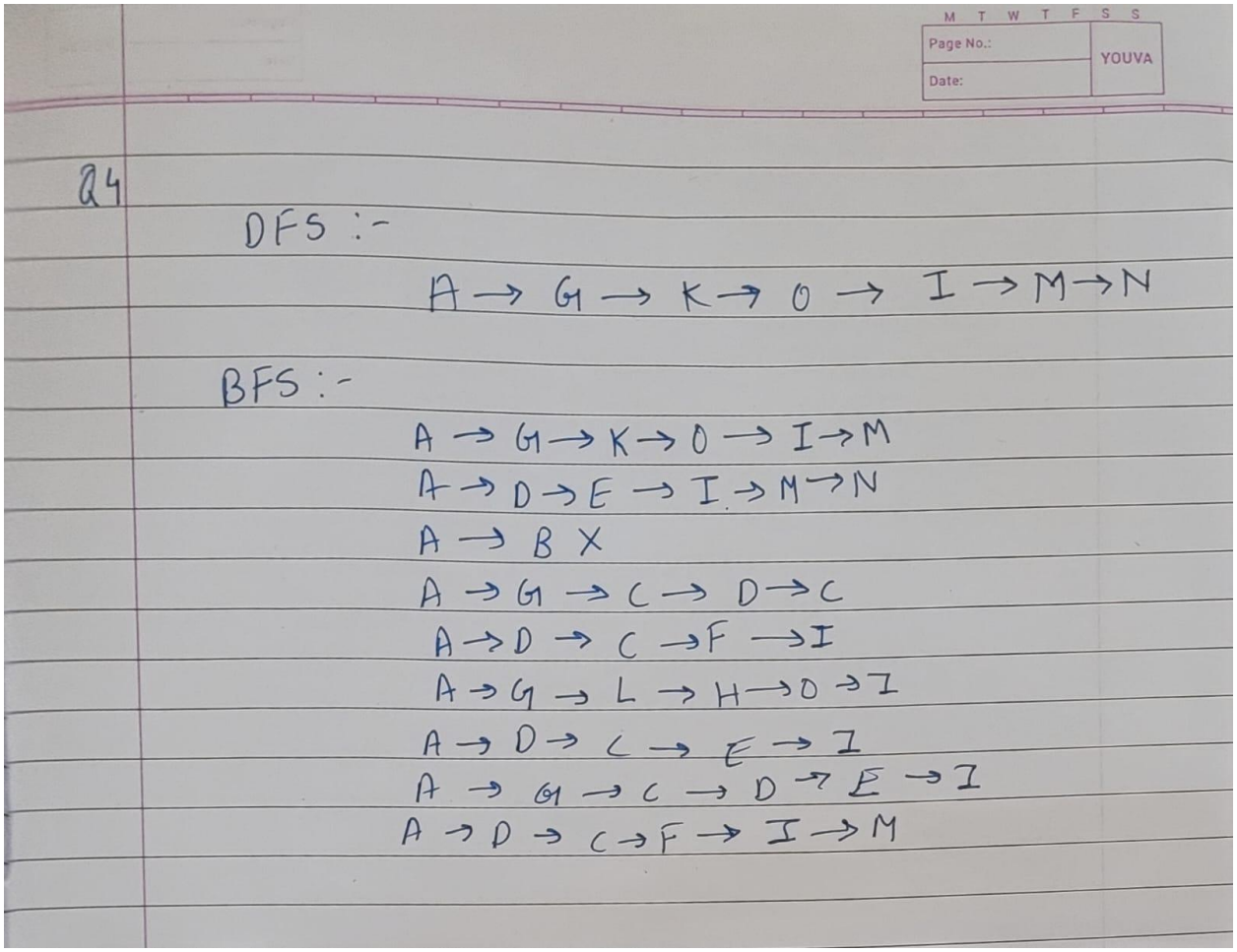
B - DFS

```
/* Editor: Omkar Ugale
DATE - 17-Aug-2022 TIME - 10:19:20*/
#include <bits/stdc++.h>

using namespace std;
const int N = 1e4 + 1;
int vis[N] = {0};
int ddfs[N];
int j = 0;
void dfs(int node, vector<int> v[])
{
    cout << node << " ";
    vis[node] = 1;
    ddfs[j++] = node;
    for (auto it : v[node])
    {
        if (vis[it])
```

```
            continue;
        vis[it] = 1;
        dfs(it, v);
    }
}
int main()
{
    int n, m;
    cin >> n >> m;
    vector<int> v[n + 1];
    for (int i = 0; i < m; i++)
    {
        int a, b;
        cin >> a >> b;
        v[a].push_back(b);
        v[b].push_back(a);
    }
    for (int i = 0; i <= n; i++)
    {
        cout << i << " : ";
        for (auto it : v[i])
            cout << it << " ";
        cout << endl;
    }
    dfs(1, v);
    for (int i = 0; i <= n; i++)
        cout << ddfs[i] << " ";
    cout << endl;
    return 0;
}
// 8 8
// 1 2
// 1 3
// 2 5
// 2 6
// 3 4
// 3 7
// 7 8
```

```
// 4 8
```



Q5)

A - DFS

```
/* Editor: Omkar Ugale
DATE - 17-Aug-2022 TIME - 07:48:42*/
#include <bits/stdc++.h>

using namespace std;
const int N = 1e4 + 1;
int vis[N] = {0};
// int level[N] = {0};
int bbfs[N];
void bfs(int node, vector<int> v[])
```

```
{
    queue<int> q;
    q.push(node);
    vis[node] = 1;
    // level[node]++;
    int i = 0;
    while (!q.empty())
    {

        int t = q.front();
        bbfs[i++] = t;
        q.pop();
        for (auto it : v[t])
        {
            if (!vis[it])
            {
                vis[it] = 1;
                q.push(it);
                // level[it] = level[t] + 1;
            }
        }
    }
}
int main()
{
    int n, m;
    cin >> n >> m;
    vector<int> v[n + 1];
    for (int i = 0; i < m; i++)
    {
        int a, b;
        cin >> a >> b;
        v[a].push_back(b);
        v[b].push_back(a);
    }
    bfs(1, v);
    for (int i = 0; i <= n; i++)
        cout << bbfs[i] << " ";
```

```
    cout << endl;
    // for (int i = 1; i <= n; i++)
    //     cout <<i<<" "<< level[i] << ", ";
    // cout << endl;

    return 0;
}
```

B - DFS

```
/* Editor: Omkar Ugale
DATE - 17-Aug-2022 TIME - 10:19:20*/
#include <bits/stdc++.h>

using namespace std;
const int N = 1e4 + 1;
int vis[N] = {0};
int ddfs[N];
int j = 0;
void dfs(int node, vector<int> v[])
{
    cout << node << " ";
    vis[node] = 1;
    ddfs[j++] = node;
    for (auto it : v[node])
    {
        if (vis[it])
            continue;
        vis[it] = 1;
        dfs(it, v);
    }
}
int main()
{
    int n, m;
    cin >> n >> m;
    vector<int> v[n + 1];
    for (int i = 0; i < m; i++)
    {
```

```
        int a, b;
        cin >> a >> b;
        v[a].push_back(b);
        v[b].push_back(a);
    }
    for (int i = 0; i <= n; i++)
    {
        cout << i << " : ";
        for (auto it : v[i])
            cout << it << " ";
        cout << endl;
    }
    dfs(1, v);
    for (int i = 0; i <= n; i++)
        cout << ddfs[i] << " ";
    cout << endl;
    return 0;
}
// 8 8
// 1 2
// 1 3
// 2 5
// 2 6
// 3 4
// 3 7
// 7 8
// 4 8
```