

# LAB-13

Name: K V Jaya Harsha

Roll no: CS23B1034

Date: 30-10-2024

Q1. BFS (in cpp)

```
// K V Jaya Harsha
// CS23B1034
#include <iostream>
#include <vector>
#include <queue>
using namespace std;

int main()
{
    int n, m;
    cin >> n >> m;

    vector<bool> visited(n + 1, false);
    vector<vector<int>> adj(n + 1);

    char x, y;
    for (int i = 0; i < m; i++)
    {
        cin >> x >> y;
        int a = x - 'A';
        int b = y - 'A';
        adj[a].push_back(b);
        adj[b].push_back(a);
    }

    queue<char> q;
    q.push(0);
    visited[0] = true;

    while (!q.empty())
    {
        int node = q.front();
        q.pop();
        cout << char(node + 'A') << endl;

        for (int neighbour : adj[node])
        {
            if (!visited[neighbour])
            {
                visited[neighbour] = true;
                q.push(neighbour);
            }
        }
    }

    return 0;
}
```

Output (q1)

```
-13\'' ; 14
```

```
10 21
```

```
A B
```

```
A C
```

```
A D
```

```
B C
```

```
B E
```

```
C E
```

```
C F
```

```
D C
```

```
D F
```

```
D G
```

```
E H
```

```
F E
```

```
F H
```

```
F J
```

```
F I
```

```
F G
```

```
G H
```

```
H J
```

```
H I
```

```
I J
```

```
A
```

```
B
```

```
C
```

```
D
```

```
E
```

```
F
```

```
G
```

```
H
```

```
J
```

```
I
```

-----

Q2. DFS (in cpp)

```
1 // K V Jaya Harsha
2 // CS23B1034
3 #include <iostream>
4 #include <vector>
5 #include <stack>
6 using namespace std;
7
8 int main()
9 {
10     int n, m;
11     cin >> n >> m;
12
13     vector<bool> visited(n, false);
14     vector<vector<int>> adj(n);
15
16     char x, y;
17     for (int i = 0; i < m; i++)
18     {
19         cin >> x >> y;
20         int a = x - 'A';
21         int b = y - 'A';
22         adj[a].push_back(b);
23         adj[b].push_back(a);
24     }
25
26     stack<int> s;
27     s.push(0);
28     visited[0] = true;
29
30     while (!s.empty())
31     {
32         int node = s.top();
33         s.pop();
34         cout << char(node + 'A') << endl;
35
36         for (int i = 0; i < adj[node].size(); i++)
37         {
38             int neighbour = adj[node][i];
39             if (!visited[neighbour])
40             {
41                 visited[neighbour] = true;
42                 s.push(neighbour);
43             }
44         }
45     }
46
47     return 0;
48 }
```

Output (q2)

```
10 21
A B
A C
A D
B C
B E
C E
C F
D F
D G
D C
E H
F E
F G
F H
F I
F J
G H
G I
H I
H J
I J
A
B
E
H
J
I
C
F
D
G
D F G H I J
```