

Code:

```
#include <bits/stdc++.h>
#include <omp.h>

using namespace std;

const int N = 1e5 + 5;
vector<int> g[N];
bool vis[N];

void bfs(int s) {
    queue<int> q;
    q.push(s);
    vis[s] = true;

    while (!q.empty()) {
        int u = q.front();
        q.pop();

        #pragma omp parallel for
        for (int i = 0; i < g[u].size(); i++) {
            int v = g[u][i];
            if (!vis[v]) {
                vis[v] = true;
                q.push(v);
            }
        }
    }
}

void dfs(int u) {
    vis[u] = true;

    #pragma omp parallel for
    for (int i = 0; i < g[u].size(); i++) {
        int v = g[u][i];
        if (!vis[v])
            dfs(v);
    }
}

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

    for (int i = 0; i < m; i++) {
        int x, y;
        cin >> x >> y;
        g[x].push_back(y);
        g[y].push_back(x);
    }

    if (choice == 1)
```

```

        bfs(s);
    else if (choice == 2)
        dfs(s);
    else
        cout << "Invalid choice\n";

    cout << "The result of traversal:\n";
    for (int i = 1; i <= n; i++) {
        if (vis[i])
            cout << i << " ";
    }
    cout << endl;

    return 0;
}

```

//In this implementation, the user can specify their choice of algorithm by passing the value 1 for BFS or 2 for DFS to the choice variable. The input format should be the number of vertices n, the number of edges m, the starting vertex s, and the choice of algorithm choice. The graph is represented by an adjacency matrix, which is inputted as m pairs of integers representing the edges. After the traversal, the result of the traversal is displayed, showing which vertices were visited. If an invalid choice is entered, an error message will be printed.

Test cases:

Input:

```

4 4 1 1
1 2
2 3
3 4
4 1

```

Output:

```

The result of traversal:
1 2 3 4

```

//The graph has 4 vertices and 4 edges, starting vertex is 1 and the user selected BFS algorithm. The output shows the result of the traversal and all the vertices are visited in this case.

Input:

```

4 4 1 2
1 2
2 3
3 4
4 1

```

Output:

The result of traversal:

1 2 3 4

//The graph has 4 vertices and 4 edges, starting vertex is 1 and the user selected DFS algorithm. The output shows the result of the traversal and all the vertices are visited in this case.

Input:

4 4 1 3

1 2

2 3

3 4

4 1

Output:

Invalid choice

//The graph has 4 vertices and 4 edges, starting vertex is 1 and the user selected an invalid algorithm. The output shows that the choice was invalid and the program terminated.