



Lab Report-03

(Breadth_First_Search)

**CSE-2212 (Design and Analysis of
Algorithms Lab)**

Submitted By:

Name: Eyasir Ahamed
Exam Roll: 413
Class Roll: 15
Registration No:
202004017

Submitted To:

Sharad Hasan
Ex. Lecturer
Dept. of CSE
Sheikh Hasina University,
Netrokona

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SHEIKH HASINA UNIVERSITY
NETROKONA, BANGLADESH

#3_Breadth First Search (BFS)

Problem Definition

Given a graph represented as an adjacency list and a starting vertex start, the problem is to traverse the graph using Breadth First Search algorithm starting from the vertex start.

Formal Statement of the Algorithm

- Initialize a queue and enqueue the starting vertex start.
- Initialize a visited array to keep track of visited vertices and mark start as visited.
- While the queue is not empty:
 - Dequeue a vertex v.
 - Visit vertex v.
 - For each adjacent vertex u of v:
 - If u is not visited, mark it as visited and enqueue u.
- Repeat step 3 until the queue becomes empty.

Complexity Analysis

- Time Complexity:
 - $O(V + E)$, where V is the number of vertices and E is the number of edges.
 - Each vertex and edge is visited once, so the time complexity is linear in terms of both vertices and edges.
- Space Complexity:

- $O(V)$ for the visited array.
- $O(V)$ for the queue in the worst case when all vertices are enqueued.
- Total: $O(V + V) = O(V)$, where V is the number of vertices.
- Additional space is required for maintaining the queue and the visited array.

Actual Code and Output

```
1  #include <iostream>
2  #include <vector>
3  #include <queue>
4
5  using namespace std;
6
7  void bfs(vector<vector<int>>& graph, int start) {
8      int n = graph.size();
9      vector<bool> visited(n, false);
10     queue<int> q;
11
12     q.push(start);
13     visited[start] = true;
14
15     while (!q.empty()) {
16         int v = q.front();
17         q.pop();
18         cout << v << " ";
19
20         for (int u : graph[v]) {
21             if (!visited[u]) {
22                 visited[u] = true;
23                 q.push(u);
24             }
25         }
26     }
27 }
28
29 int main() {
30     vector<vector<int>> graph = {
31         {1, 2},
32         {0, 3, 4},
33         {0, 4},
34         {1},
35         {1, 2}
36     };
37
38     cout << "BFS traversal starting from vertex 0: ";
39     bfs(graph, 0);
40     cout << endl;
41
42     return 0;
43 }
```

```
BFS traversal starting from vertex 0: 0 1 2 3 4
[Finished in 437ms]
```