

Graphs - Breadth-First Search (BFS)

Breadth-First Search (or Breadth-First Traversal) is one of the ways to traverse a graph. It takes its name after the fact that it expands the frontier between discovered and undiscovered vertices uniformly across the breadth of the frontier. This in turn implies that the algorithm will first visit all the vertices connected to "u" at a distance "k" before discovering the vertices at a distance "k+1" from "u". The algorithm explained in:

[1] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., 2009. Introduction to algorithms. MIT press.

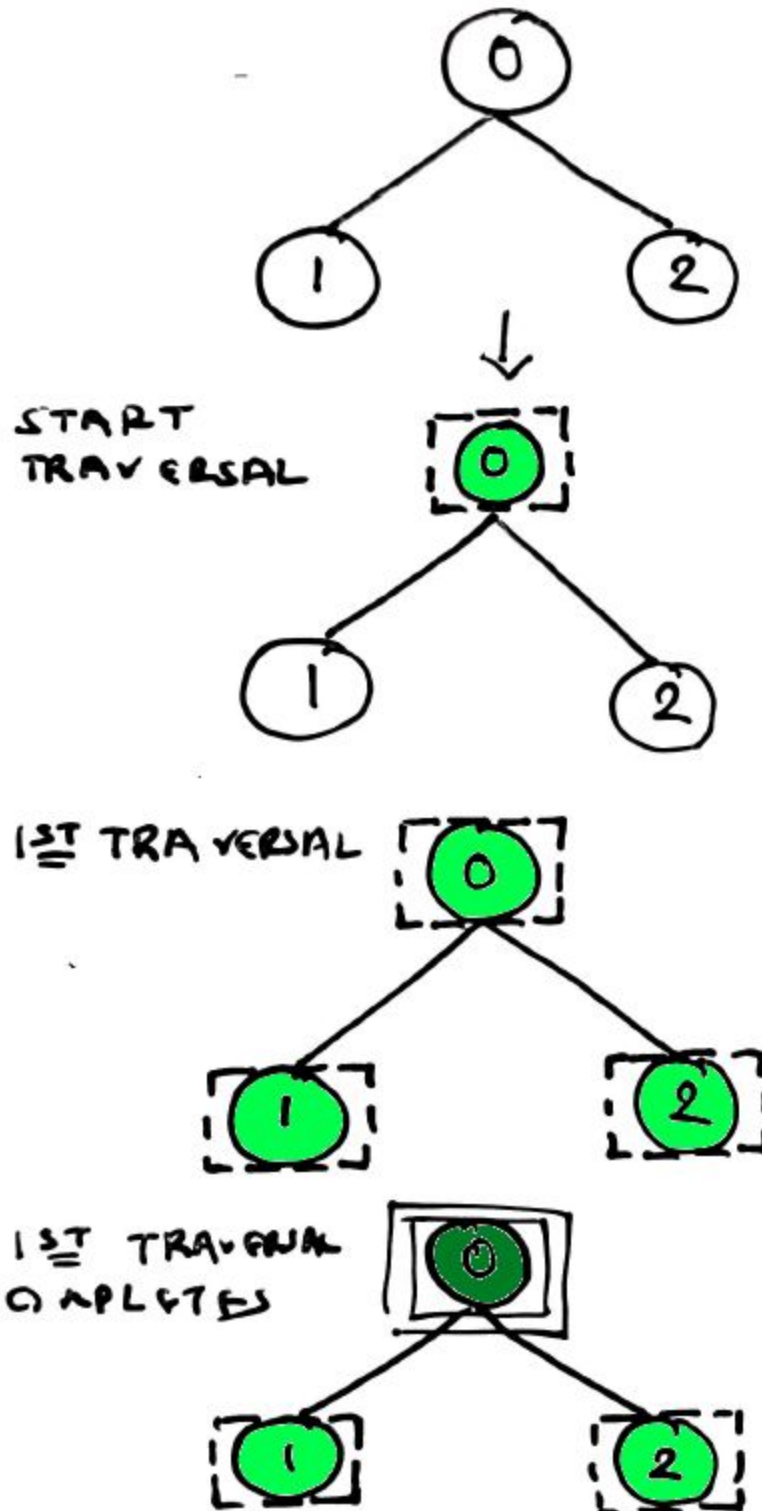
employs the concept of "color" to check if a vertex is discovered fully or partially or if it is undiscovered. It further keeps track of the distance that a vertex "u" is from the source vertex "s".

In pseudocode we can write:

Algorithm 1. Breadth-First Search Pseudocode as in [1]

```
BFS(G, s)
1  for each vertex u in G.V - {s}
2      u.color = white
3      u.d = INF
4      u.p = NIL
5  s.color = green
6  s.d = 0
7  s.p = NIL
8  Q = NULL
9  ENQUEUE(Q, s)
10 while Q != NULL
11     u = DEQUEUE(Q)
12     for each v in G.Adj[u]
13         if v.color == white
14             v.color = green
15             v.d = u.d + 1
16             v.p = u
17             ENQUEUE(Q, v)
18     u.color = dark_green
```

The following drawing visualizes steps of a Breadth-First Search.



Task: Implement BFS on the basis of the Pseudocode in Algorithm 1.

For assistance in this implementation, please consider the following "driving" program.

```

// Driver Function
int main()
{
    // Create graph with 7 nodes and 6 edges
    int n = 7;

    // the graph vector for its nodes (vertices)
    vector <int> g[n];

    addEdge(g, 0, 1);
    addEdge(g, 0, 2);
    addEdge(g, 1, 3);
    addEdge(g, 1, 4);
    addEdge(g, 2, 5);
    addEdge(g, 2, 6);

    BFSFull(g, n);

    return 0;
}

```

You are requested to implement the methods

- `void addEdge(vector <int> g[], int u, int v)`
- `void BFSFull(vector <int> g[], int n)`

Reference: <https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/>