

Batch: D3 Roll No.: 16010123294

Experiment / assignment / tutorial No. 08

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: Study of Graph traversal methods

Objective: To Understand Graph Traversal Methods - BFS & DFS.

Expected Outcome of Experiment:

| CO | Outcome |
|----|--|
| 2 | Apply linear and non-linear data structure in application development. |

Books/ Journals/ Websites referred:

1. *Fundamentals Of Data Structures In C* – Ellis Horowitz, Satraj Sahni, Susan Anderson-Fred
2. *An Introduction to data structures with applications* – Jean Paul Tremblay, Paul G. Sorenson
3. *Data Structures A Pseudo Approach with C* – Richard F. Gilberg & Behrouz A. Forouzan
4. <https://www.geeksforgeeks.org/binary-tree-data-structure/>
5. <https://www.thecrazyprogrammer.com/2015/03/c-program-for-binary-search-tree-insertion.html>

Abstract:

Graph is a non-linear data structure consisting of vertices and edges. The vertices are sometimes also referred to as nodes and the edges are lines or arcs that connect any two nodes in the graph. More formally a Graph is composed of a set of vertices(V) and a set of edges(E). The graph is denoted by G(V, E).

Graph data structures are a powerful tool for representing and analyzing complex relationships between objects or entities. They are particularly useful in fields such as social network analysis, recommendation systems, and computer networks. In the field of sports data science, graph data structures can be used to analyze and understand the dynamics of team performance and player interactions on the field.

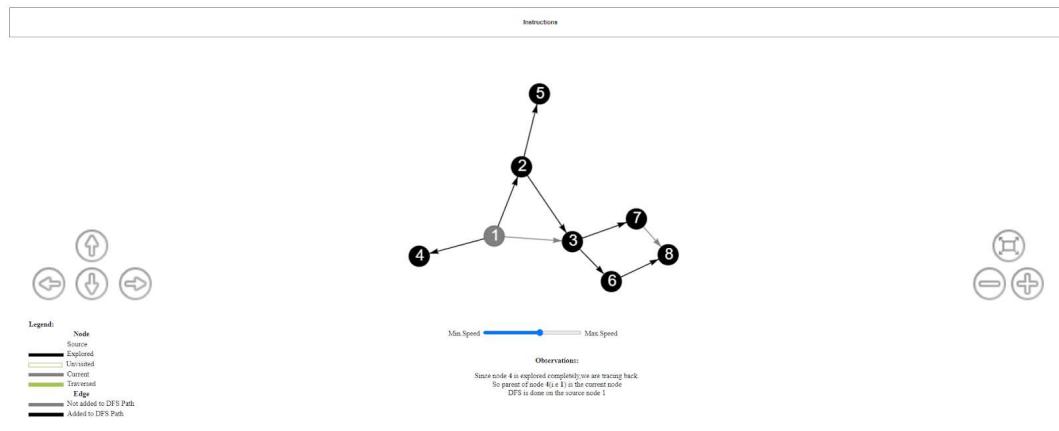
BFS & DFS on the assigned graphs:

DFS

Computer Science and Engineering > Data Structures - 1 > Experiments

| | |
|--|---|
| <p>Aim</p> <p>Overview</p> <p>Basics of Graphs</p> <p>Graph Traversals</p> <p>Pretest</p> <p>Depth First Search</p> <p>Code Assessment</p> <p>Analysis</p> <p>Posttest</p> <p>Further Readings/References</p> <p>Feedback</p> | <p>1. Which one of the following is an application of a directed graph? <input type="radio"/> A Mobile communication graph <input checked="" type="radio"/> B Water network graph <input type="radio"/> C Scheduling courses with prerequisites <input type="radio"/> D Graph of friends</p> <p>2. If m & n represent the number of vertices & edges respectively, then which one of the following can not be true. <input type="radio"/> A $m = 0 \& n = 0$ <input type="radio"/> B $m = 0 \& n = 0$ <input checked="" type="radio"/> C $m = 0 \& n = 0$ <input type="radio"/> D $m = 10000 \& n = 1$</p> <p>3. Which one of the following data structures supports random access of elements? <input type="radio"/> A Linked List <input type="radio"/> B Tree <input checked="" type="radio"/> C Array <input type="radio"/> D Graph</p> <p>4. In which of the following data structures, you can traverse in only one direction <input type="radio"/> A Undirected graph <input type="radio"/> B Doubly linked list <input type="radio"/> C Array <input checked="" type="radio"/> D Directed Graph</p> <p>5. Which one of the following is an application of a directed graph? <input type="radio"/> A Undirected graph <input type="radio"/> B Scheduling courses with prerequisites <input type="radio"/> C Data broadcasting from parent to children nodes <input type="radio"/> D Family tree <input checked="" type="radio"/> E All of the above</p> <p>Submit Quiz</p> |
|--|---|

Instructions



Legend:

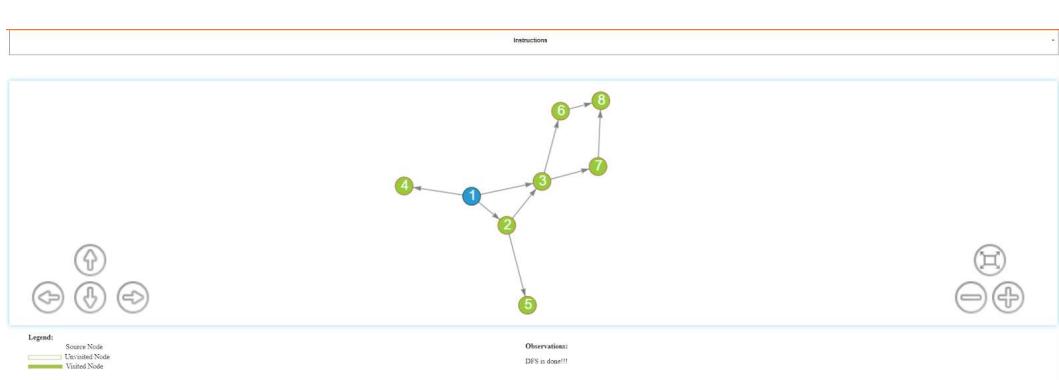
- Node
- Source
- Exploded
- Unselected
- Current
- Visited
- Edge
- Not added to DFS Path
- Added to DFS Path

Observations:

Since node 4 is explored completely we are tracing back.
So parent of node 4 is 1 is the current node.
DFS is done to the source node 1.

Min Speed Max Speed

Instructions



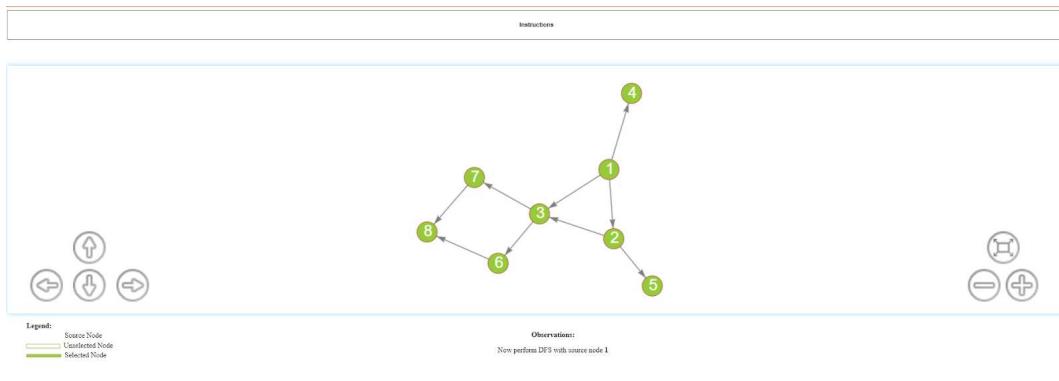
Legend:

- Source Node
- Unselected Node
- Visited Node

Observations:

DFS is done!!!

Instructions



Legend:

- Source Node
- Unselected Node
- Selected Node

Observations:

Now perform DFS with source node 1.

Reset New graph Home Previous Next

Computer Science and Engineering > Data Structures - 1 > Experiments

Aim
 Overview
 Basics of Graphs
 Graph Traversals
 Pretest
 Depth First Search ▾
 Aim
 Concept
 Algorithm
 Demo
 Practice
 Exercise
 Quiz
 Code Assessment
 Analysis ▾
 Posttest
 Further Readings/References
 Feedback

1. Which one of the following data structures is used in DFS?
 A) Heap
 B) Dequeue
 C) Linked List
 D) Stack
2. What is backtracking in DFS?
 A) Going to the sibling node when all children nodes have been visited
 B) Going to the parent node when a child node has been visited
 C) Going to the sibling node when all the children nodes have been visited
 D) Going to the sibling node when a child node has been visited

Submit Quiz

2 out of 2

Depth First Search

Computer Science and Engineering > Data Structures - 1 > Experiments

Aim
 Overview
 Basics of Graphs
 Graph Traversals
 Pretest
 Depth First Search ▾
 Code Assessment
 Analysis ▾
 Aim
 Space and Time Complexity
 Comparison of BFS with DFS
 Quiz
 Posttest
 Further Readings/References
 Feedback

1. If there are 10 edges in a graph, in the worst case how many edges can be traversed?
 A) 10
 B) 1
 C) 5
 D) 100
2. Which one of the following is correct?
 A) DFS uses queue & BFS can be done using stack & recursion
 B) In DFS, all the neighbors are traversed before other nodes
 C) DFS is a vertex-based algorithm while BFS is an edge-based algorithm
 D) BFS is an optimal algorithm while DFS is not optimal

Submit Quiz

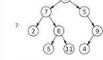
2 out of 2

Depth First Search

Computer Science and Engineering > Data Structures - 1 > Experiments

Aim
 Overview
 Basics of Graphs
 Graph Traversals
 Pretest
 Depth First Search ▾
 Code Assessment
 Analysis ▾
 Aim
 Further Readings/References
 Feedback

1. Which one of the following steps is incorrect when performing DFS?
 A) DFS is for picking any one of the greatest vertices on top of a vertex stack as source node of DFS
 B) In DFS, all the neighbors are traversed before other nodes
 C) In DFS, all the neighbors are traversed after other nodes
 D) DFS is a vertex-expanding search and it finds the solution quickly
2. What is the time complexity of DFS? V is the number of vertices & E is the number of edges
 A) O(V)
 B) O(E)
 C) O(V+E)
 D) O(V+E)
3. Pick the incorrect option.
 A) DFS uses stack or recursion
 B) BFS uses queue & recursion
 C) BFS uses queue & recursion while the node has no children
 D) BFS uses breadth first search while other nodes
4. Which one of the following are applications of DFS?
 A) Nearest neighbor
 B) Minimum Spanning Tree
 C) Shortest path algorithms
 D) All of the above
5. DFS can only be applied on Trees & not on Graphs.
 A) True
 B) False
6. Pick the incorrect option.
 A) DFS uses stack or recursion along with before exploring other options
 B) BFS uses stack or recursion
 C) DFS uses stack or recursion
 D) BFS uses breadth first search



When we apply DFS on the above graph, which one of the following order of traversal is not possible?

- A) 1234567
 B) 1235467
 C) 123454321
 D) 1234567654321

Submit Quiz

Depth First Search

BFS

Aim
Overview
Basics of Graphs
Graph Traversal
Basics of Queues
Pretest
BFS ▾
Analysis ▾
Posttest
Further Readings/References
Feedback

Choose difficulty:

Beginner

Intermediate

Advanced

1 Which of the following policies does a queue follow?

a) FIFO - First In First Out Explanation

b) LIFO - Last In First Out Explanation

c) FILO - First In Last Out Explanation

d) Random order Explanation

2 Which of the following describes a standard graph traversal algorithm?

a) Visiting all the edges of the graph Explanation

b) Visiting all the vertices of the graph Explanation

c) Detecting all the cycles in the graph Explanation

d) None of the above Explanation

3 Consider the following undirected graph:

Vertices: V = {a, b, c, d, e, f}

Edges: E = {{a, b}, {a, c}, {b, d}, {b, e}, {c, d}, {c, e}, {d, f}}

Where each array within E signifies an edge between the two mentioned vertices.

Which of the following data structures is represented by the above graph?

a) Tree Explanation

b) Cycle graph Explanation

c) Disconnected Graph Explanation

d) Complete Graph Explanation

4 Consider the following undirected graph:

Vertices, V = {a, b, c, d, e, f}

Edges: E = {{a, b}, {a, c}, {b, d}, {b, e}, {c, d}, {c, e}, {d, f}}

Where each array within E signifies an edge between the two mentioned vertices.

If we were to store this graph's vertices in a queue in the order top to bottom(parent to child) and left to right(edges that appear first in the edge matrix appear first in the queue) with 'a' as the root, what index would vertex 'e' be stored at (assume 0 indexing for the queue and no deletions)?

a) 2

b) 3 Explanation

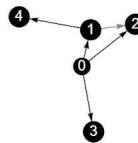
c) 4 Explanation

d) 5 Explanation

[Show Answer](#)

Breadth First Search

Instructions



Legend:
■ Node
■ Explode
■ Unvisited
■ Visited
■ Edge
■ Not added to DFS Path
■ Added to DFS Path

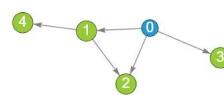
Observations:

BFS is done on the node 0 !!!

Sequence of nodes visited in performing BFS on node 0: 0, 1, 2, 3, 4

Min Speed Max Speed

Instructions



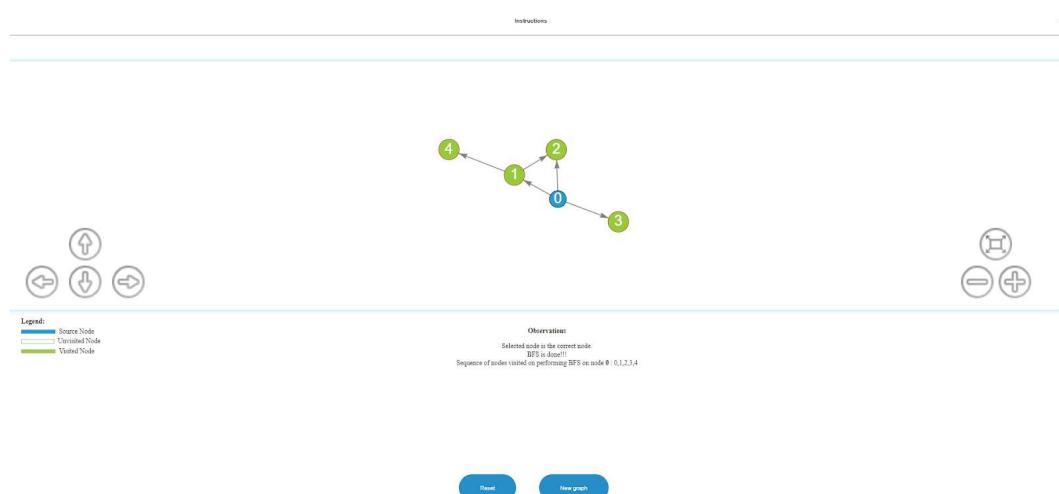
Legend:
■ Source Node
■ Unvisited Node
■ Visited Node

Observations:

Your sequence: 0, 1, 2, 3, 4

Correct!

Reset New search Submit



Computer Science and Engineering > Data Structures - 1 > Experiments

[Aim](#)
[Overview](#)
[Basics of Graphs](#)
[Graph Traversal](#)
[Basics of Queues](#)
[Pretest](#)
BFS ▾
[Aim](#)
[Concept](#)
[Algorithm](#)
[Demo](#)
[Practice](#)
[Exercise](#)
[Quiz](#)
Analysis ▾
[Posttest](#)
[Further Readings/References](#)
[Feedback](#)

Breadth First Search

- Choose difficulty: Beginner Intermediate
1. Which of the following are applications of BFS?
 a To find the Minimum Spanning Tree (MST) Explanation
 b To find the shortest distance to a node from the root Explanation
 c Cycle detection in undirected graphs Explanation
 d All of the above Explanation
2. Imagine a 4x4 grid (assume 1 indexing) where the hero stands in the cell (2,3) and the villain is in the cell (3,1). Assuming the hero can only move up, down, left and/or right to the immediately adjacent cells and cannot move out of the grid, how many steps will it take for the hero to reach the villain (given that the villain is stationary)?
 a 1 Explanation
 b 2 Explanation
 c 3 Explanation
 d 4 Explanation
- [Submit Quiz](#)
Score: 2 out of 2

Computer Science and Engineering > Data Structures - 1 > Experiments

[Aim](#)
[Overview](#)
[Basics of Graphs](#)
[Graph Traversal](#)
[Basics of Queues](#)
[Pretest](#)
BFS ▾
[Analysis](#) ▾
[Aim](#)
[Time and Space Complexity](#)
[Comparison with other Algorithms](#)
[Quiz](#)
[Posttest](#)
[Further Readings/References](#)
[Feedback](#)

Breadth First Search

1. Consider the following graph:
 Vertices: V = {a, b, c, d, e, f}
 Edges: E = {a, b}, {a, c}, {b, d}, {b, e}, {c, e}, {d, f}, {e, f}
- Where each entry within E signifies an edge between the two mentioned vertices.
- How many iterations of the queue would it take for the algorithm to traverse this graph completely?
- a 3 Explanation
 b 5 Explanation
 c 6 Explanation
 d 7 Explanation
2. When will the space complexity of BFS be greater than DFS? Note that maximum height in the options refers to the longest thread of vertices from the root to a leaf or final non-repeating vertex.
 a If the maximum height is less than the maximum number of nodes in a single level Explanation
 b If the maximum height is greater than the maximum number of nodes in a single level Explanation
 c BFS and DFS have same the space complexity Explanation
 d Space complexity of DFS is always greater than that of BFS Explanation
- [Submit Quiz](#)
Score: 2 out of 2



Computer Science and Engineering > Data Structures - 1 > Experiments

Aim
Overview
Basics of Graphs
Graph Traversal
Basics of Queues
Pretest
BFS ▾
Analysis ▾
Posttest
Further Readings/References
Feedback

Breadth First Search

- Choose difficulty: Intermediate Advanced
- 1 Which of the following is a use of the extra 'visited' array (the array used to keep track of which nodes have been visited/traversed) in BFS?
 To avoid getting stuck in a cycle. Explanation
 To decide which node to traverse next. Explanation
 To preemptively end the algorithm when all nodes are marked as visited thus saving time. Explanation
 None of the above. Explanation
- 2 What would happen if we used a stack instead of a queue in BFS?
 The algorithm would simply traverse the graph in the reverse order i.e. from bottom to top (leaves to root). Explanation
 The algorithm would become equivalent to DFS. Explanation
 The algorithm would not work properly, i.e. it will not traverse the graph properly and/or completely. Explanation
 No change in the algorithm, i.e. it remains unaffected. Explanation
- 3 Why is the time complexity of BFS $O(V + E)$?
 Because it considers all vertices and edges in the worst case. Explanation
 Because it explores all edges in a vertex. Explanation
 This is not the correct time complexity of BFS. Explanation
 None of the above. Explanation
- 4 Consider the following graph:
Vertices: V = {a, b, c, d, e, f}
Edges: E = {a, b, b, c, b, d, b, a, c, e, f, c, f}
Where each entry within E signifies an edge between the two mentioned vertices and a is the root.
Which of the following represents the correct sequence of the queue used in BFS to traverse the above graph?
 a → {a} → {b, c} → {b, f} → {b, e} → {d} → {d} → {e, f, h} Explanation
 b → {a} → {b, c} → {b, d} → {c, d} → {a, b, c, d, f, h} Explanation
 c → {a} → {b, c} → {c, d} → {d} → {e, f} → {f} → {g, h} Explanation
 d → {a} → {b, c} → {c, d} → {d} → {e, f} → {f} → {g, h} Explanation

Score: 4 out of 4

Conclusion:-We implement the BFS and DFS via virtual lab for deeper understanding.

PostLab Questions:

1) Differentiate between BFS & DFS

Traversal Approach:

BFS: Explores neighbors level by level, moving outward from the starting node.

DFS: Explores as far down a branch as possible before backtracking.

Data Structure:

BFS: Uses a queue to manage nodes, ensuring a level-order traversal.

DFS: Uses a stack (or recursion) to go deep into each branch before returning.

Time Complexity:

Both BFS and DFS have a time complexity of $O(V + E)$, where V is the number of vertices and E is the number of edges.

Applications:

BFS: Ideal for finding the shortest path in unweighted graphs.

DFS: Useful for cycle detection, pathfinding in mazes, and topological sorting.

2) Applications of BFS & DFS

BFS Applications:

- Shortest Path in Unweighted Graphs: Finds the shortest path between two nodes in an unweighted graph.
- Web Crawlers: Crawls web pages level by level.
- Social Networking Sites: Finds connections or friends within a certain degree.
- GPS Navigation: Helps in finding the shortest route in maps (unweighted scenarios).

DFS Applications:

- Cycle Detection: Identifies cycles in directed and undirected graphs.
- Topological Sorting: Used in scheduling tasks where certain tasks depend on others.
- Maze/Pathfinding Algorithms: Helps in exploring all possible paths and backtracking if a path is blocked.
- Solving Puzzles: Solves problems like Sudoku, where depth exploration is needed to find a solution.