

**Batch: A1      Roll No.: 16010123012**

**Experiment / assignment / tutorial No. 8**

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

**Signature of the Staff In-charge with date**

**Title: Study of Graph traversal methods**

Computer Science and Engineering > Data Structures - 1 > Experiments

### Depth First Search

**1. Which one of the following is an application of a directed graph?**

- ☐ a. Media communication graph
- ☐ b. Water network graph
- ☒ c. Scheduling courses with prerequisites
- ☐ d. Graph of friends

**2. If m & n represent the number of vertices & edges respectively, then which one of the following can not be true.**

- ☐ a.  $m = 0 \& n = 0$
- ☐ b.  $m = 0 \& n = 1$
- ☒ c.  $m = 0 \& n = 2$
- ☐ d.  $m = 10000 \& n = 1$

**3. Which one of the following data structures supports random access of elements?**

- ☐ a. Linked List
- ☐ b. Tree
- ☒ c. Array
- ☐ d. Graph

**4. In which of the following data structures, you can traverse in only one direction**

- ☐ a. Undirected graph
- ☐ b. Doubly linked list
- ☐ c. Array
- ☒ d. Directed Graph

**5. Which one of the following is an application of a directed graph?**

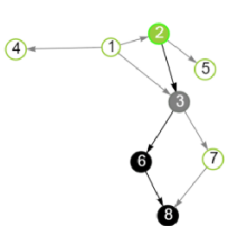
- ☐ a. Scheduling courses with prerequisites
- ☐ b. Data broadcasting from parent to children nodes
- ☐ c. Family tree
- ☒ d. All of the above

[Submit Quiz](#)  
9 out of 5

---

### Depth First Search

[Rate Me](#) [Report a Bug](#)



**Legend:**

**Node**

- Source
- Explored
- Unvisited
- Current
- Traversed

**Edge**

- Not added to DFS Path
- Added to DFS Path

Min.Speed  Max.Speed

**Observations:**

Since node 6 is explored completely, we are tracing back.  
So parent of node 6 (i.e 3) is the current node

[Reset](#) [New graph](#) [Pause](#) [Next](#)

Computer Science and Engineering > Data Structures - I > Experiments

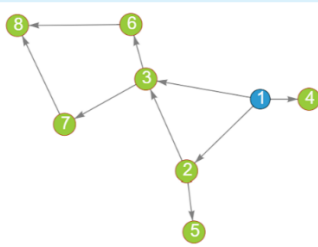
Am  
Overview  
Basics of Graphs  
Graph Traversals  
Pretest  
Depth First Search  
Code Assessment  
Analysis  
Posttest  
Further Readings/References  
Feedback

## Depth First Search

- Which one of the following is an application of a directed graph?
    - ☐ a. Mobile communication graph
    - ☐ b. Water network graph
    - ☒ c. Scheduling courses with prerequisites
    - ☐ d. Graph of friends
  - If  $m$  &  $n$  represent the number of vertices & edges respectively, then which one of the following can not be true.
    - ☐ a.  $m = 0$  &  $n = 0$
    - ☐ b.  $m = 0$  &  $n = 6$
    - ☒ c.  $m = 6$  &  $n = 0$
    - ☐ d.  $m = 10000$  &  $n = 1$
  - Which one of the following data structures supports random access of elements?
    - ☐ a. Linked List
    - ☐ b. Tree
    - ☒ c. Array
    - ☐ d. Graph
  - In which of the following data structures, you can traverse in only one direction.
    - ☐ a. Undirected graph
    - ☐ b. Doubly linked list
    - ☐ c. Array
    - ☒ d. Directed Graph
  - Which one of the following is an application of a directed graph?
    - ☐ a. Scheduling courses with prerequisites
    - ☐ b. Data broadcasting from parent to children nodes
    - ☐ c. Family tree
    - ☒ d. Job of the person
- [Submit Quiz](#)  
5 out of 5

Depth First Search

Instructions



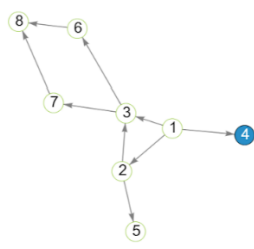
Legend: Source Node (blue), Unvisited Node (green), Visited Node (red)

Observations: DFS is done!!!

[Reset](#) [New graph](#)

Depth First Search

Instructions




Legend: Source Node (blue), Unvisited Node (green), Selected Node (red)

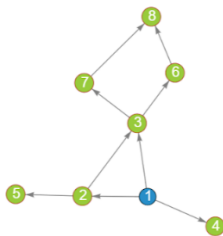
Observations: Your urgency : 4 Correct!

[Reset](#) [New graph](#) [Submit](#)


Instructions




**Legend:**  
● Source Node  
● Unselected Node  
● Selected Node



**Observations:**  
 Your sequence : 1,2,3,6,8,7,5,4  
 Correct!



[Reset](#) [New graph](#) [Submit](#)



[Rate Me](#) [Report a Bug](#)


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

## Depth First Search

- Which one of the following data structures is used in DFS?  
☐ a) Heap  
☐ b) Dequeue  
☐ c) Linked List  
☒ d) Stack
- What is backtracking in DFS?  
☒ a) Going to the parent node if all the children nodes have been visited  
☐ b) Going to the parent node when a children 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 children node has been visited

[Submit Quiz](#)  
 2 out of 2



[★★★★☆](#) [Rate Me](#) [Report a Bug](#)

- Basics of Graphs
- Graph Traversals
- Pretest
- Depth First Search ▼
- Code Assessment
- Analysis ▼
- Posttest
- Further Readings/References
- Feedback

PROBLEM 1
PROBLEM 2

### Basic Depth First Search

Complete

You are given the list of edges of a directed graph and the root node. Perform depth first search to find the correct order of visitation of nodes. You are advised to first construct a graph from the edge list then proceed further.

**Input Format**  
 Input consists of an array of edges as [node1, node2]. For node1 as parent. The first of the edges denote the edge with the left child. other latter edge with the right child.

**Output Format**  
 An array of nodes in the correct order of visitation

```

1  JS
2  graph = new Map();
3  const [parent, child] = edge;
4  if (!graph[parent]) {
5    graph[parent] = [];
6  }
7  graph[parent].push(child);
8  });
9
10
11
12  const visited = new Set();
13  const result = [];
14
15  const dfs = (node) => {
16    if (visited.has(node)) return;
17    visited.add(node);
18    result.push(node);
19  }
                
```

Output

✓ Correct

Code Output:

[1,2,3,4,5,6]


Expected Output: ☒

Submit

Department of Computer Engineering

DS Sem-III – July-Dec 2024

Page -



★★★★☆
[Rate Me](#)
[Report a Bug](#)

---

**Basics of Graphs**

Graph Traversals

Pretest

Depth First Search ▾

**Code Assessment**

Analysis ▾

Posttest

Further Readings/References

Feedback

PROBLEM 1

PROBLEM 2

### Detect Cycle in Directed Graph

Intermediate

Implement DFS to detect cycles in a directed graph.

**Input Format**

Input consists of a directed graph represented as an array of edges, where each edge [parent, child] denotes a directed connection from parent to child, and a starting vertex.

**Output Format**

Boolean indicating whether the graph contains a cycle.

```

1 JS
2
3 const visited = new Set();
4
5 const recStack = new Set();
6
7
8 const dfs = (node) => {
9   if (!graph.has(node)) return false;
10  if (recStack.has(node)) return true;
11  if (visited.has(node)) return false;
12
13  visited.add(node);
14  recStack.add(node);
15
16  for (const neighbor of graph.get(node)) {
17    if (dfs(neighbor)) {
18      return true;
19    }
20  }
21  return false;
                
```


#### Output

Code Output:

Submit

[Community Links](#)
[Contact Us](#)
[Follow Us](#)

Go to Settings to activate Windows.



Phone: General information 011-26582960

⏮ ⏪ ⏩ ⏭ 🔍 🔄

Rate Me

Report a Bug

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

Depth First Search

1. If there are 10 edges in a graph, in the worst case how many edges can be traversed?

☐ a. 0

☐ b. 1

☐ c. 5

☒ d. 10

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

The image shows two screenshots of a 'Virtual Labs' web application. The top screenshot is a quiz for Depth First Search (DFS) with 6 questions. The bottom screenshot is the 'Breadth First Search' overview page, showing difficulty levels (Beginner, Intermediate, Advanced) and a list of questions with explanations.

**DFS Quiz Questions:**

- What is the time complexity of DFS? V is the number of vertices & E is the number of edges.
  - ☐ a.  $O(V^2)$
  - ☒ b.  $O(V \cdot E)$
  - ☐ c.  $O(E^2)$
  - ☐ d.  $O(V \log V)$
- Pick the incorrect option.
  - ☐ a. DFS consumes less memory & DFS consumes less memory
  - ☐ b. DFS goes deeper & deeper until the node has no children
  - ☐ c. DFS involves traversing neighbors before other nodes
  - ☒ d. DFS uses recursion
- Which one of the following are applications of DFS?
  - ☐ a. Topological sorting
  - ☐ b. Minimum Spanning Tree
  - ☐ c. Finding connected components
  - ☒ d. All of the above
- DFS can only be applied on Trees & not on Graphs.
  - ☐ a. True
  - ☒ b. False
- Pick the incorrect option.
  - ☐ a. DFS uses exhaustive traversal along one path before exploring other options.
  - ☒ b. DFS uses stack to store the nodes visited before exploring other options.
  - ☐ c. DFS uses stack or recursion
  - ☐ d. DFS can be used in topological sorting

A graph is shown with 12 nodes and edges. The question asks: "When we apply DFS on the above graph, which one of the following order of traversal is not possible?"

**BFS Overview Page:**

Choose difficulty: ☒ Beginner ☒ Intermediate ☒ Advanced

- Which of the following policies does a queue follow?
  - ☒ a. FIFO - First In First Out
  - ☐ b. LIFO - Last In First Out
  - ☐ c. FILO - First In Last Out
  - ☐ d. Random order
- Which of the following describes a standard graph traversal algorithm?
  - ☐ a. Visiting all the edges of the graph
  - ☒ b. Visiting all the vertices of the graph
  - ☐ c. Detecting all the cycles in the graph
  - ☐ d. None of the above
- Consider the following undirected graph: Vertices: V = {a, b, c, d, e, f, g}. Edges: E = {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}. Which of the following data structures is represented by the above graph?
  - ☐ a. Tree
  - ☒ b. Cycle graph
  - ☐ c. Disconnected Graph
  - ☐ d. Complete Graph
- Consider the following undirected graph: Vertices: V = {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}. Edges: E = {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}. Which of the following data structures is represented by the above graph?
  - ☐ a. Tree
  - ☒ b. Cycle graph
  - ☐ c. Disconnected Graph
  - ☐ d. Complete Graph

## Code:

```
#include <stdio.h>

#include <stdlib.h>
int visit[20] = {0};
int v[20] = {0};
typedef struct node
{
    int data;
    struct node *prev;
    struct node *link;
} node;

typedef struct queue
{
```

```
    struct node *rr;
    struct node *fr;
} que;

int dequeue(que *q)
{
    node *temp;
    if (q->rr != NULL)
    {
        temp = q->rr;
        int d = temp->data;
        q->rr = temp->prev;
        if (q->rr != NULL)
            q->rr->link = NULL;
        else
            q->fr = NULL;
        return d;
    }
    return 0;
}

void enqueue(int ch, que *q)
{
    node *nnode;
    nnode = (node *)malloc(sizeof(node));
    nnode->data = ch;
    nnode->link = NULL;
    nnode->prev = NULL;
    if (q->fr == NULL)
    {
        q->fr = nnode;
        q->rr = nnode;
    }
    else
    {
        nnode->link = q->fr;
        q->fr->prev = nnode;
        q->fr = nnode;
    }
}

void display(que *q)
{
    node *temp;
    temp = q->fr;
    while (temp != NULL)
```

```
{
    printf(" %c", temp->data);
    temp = temp->link;
}
}

void dfs(int t, int a[20][20], int n)
{
    int i, j;
    printf("%d->", t);
    visit[t - 1] = 1;
    for (i = 0; i < n; i++)
        if (a[t - 1][i] == 1 && visit[i] == 0)
            dfs(i + 1, a, n);
}

void bfs(int t, int a[20][20], int n, que *q)
{
    int i, j;
    printf("%d->", t);
    int temp;
    enqueue(t, q);
    v[t - 1] = 1;
    while (q->fr != NULL)
    {
        temp = dequeue(q);
        for (i = 0; i < n; i++)
        {
            if (a[temp - 1][i] == 1 && v[i] == 0)
            {
                enqueue(i + 1, q);
                printf("%d->", i + 1);
                v[i] = 1;
            }
        }
    }
}

int main(void)
{
    printf("Enter number of vertices:\n");
    int n, i, j, e, p, q;
    scanf("%d", &n);
    int a[20][20];
    for (i = 0; i < n; i++)
    {
```

```
        visit[i] = 0;
        for (j = 0; j < n; j++)
            a[i][j] = 0;
    }
    printf("Enter number of edges:\n");
    scanf("%d", &e);
    printf("\nEnter 1 for undirected graph and 0 for directed graph:");
    int t;
    scanf("%d", &t);
    for (i = 0; i < e; i++)
    {
        printf("Enter edge vertex(p,q):\n");
        scanf("%d%d", &p, &q);
        a[p - 1][q - 1] = 1;
        if (t == 1)
            a[q - 1][p - 1] = 1;
    }

    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
    printf("Enter Element from where you want to start dfs and bfs:");
    int d;
    scanf("%d", &d);
    printf("\n DFS:\n");
    dfs(d, a, n);
    que q1;
    q1.fr = q1.r = NULL;
    printf("\n BFS:\n");
    bfs(d, a, n, &q1);
    return 0;
}
```

**Output:**



```
Enter number of vertices:
4
Enter number of edges:
4

Enter 1 for undirected graph and 0 for directed graph:1
Enter edge vertex(p,q):
4 3
Enter edge vertex(p,q):
7 5
Enter edge vertex(p,q):
8 6
Enter edge vertex(p,q):
9 3
0 0 0 0
0 0 0 0
0 0 0 1
0 0 1 0
Enter Element from where you want to start dfs and bfs:3

DFS:
3->4->
BFS:
3->4->
```