Data Structure
RCS-305

UNIT-4

By:

Ms.Kritika Sharma

A.P.(CSE Dept)

Department of Computer Science & Engineering

Vision

To build strong teaching environment that responds the need of industry and challenges of society.

Mission

•Developing strong mathematical & computing fundamentals among the students.
•Extending the role of computer science and engineering in diverse areas.
•Imbibing the students with a deep understanding of professional ethics and high integrity to serve the nation.
•Providing an environment to the students for their growth both as individuals and as globally competent Computer Science professional.
•Outreach activities will contribute to the overall wellbeing of society.

Presentation Outline

- Graphs
- Directed graphs
- Undirected Graphs
- Representation of Graphs
- Graph Traversal

Learning Objectives

To learn what a graph is and how it is used.

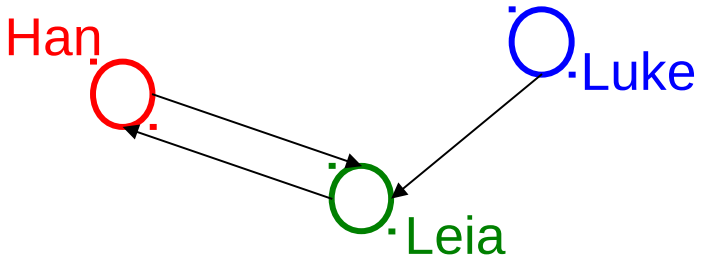
To implement the graph abstract data type using multiple internal representations.

To see how graphs can be used to solve a wide variety of problems.

To implement algorithm on minimum cost spanning trees.

What is a graph?

A graph G consists of two sets:
a finite, nonempty set of vertices V(G)
a finite, possible empty set of edges E(G) .
G(V,E) represents a graph

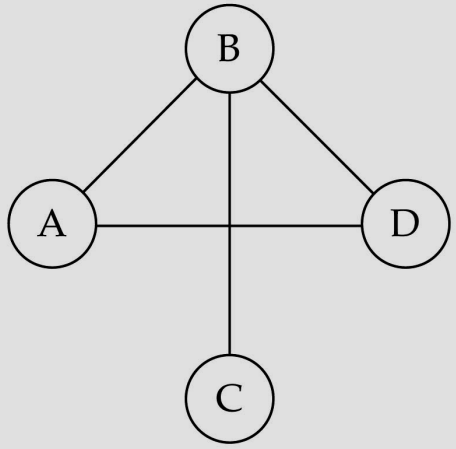



```
V = {Han, Leia, Luke}
E = {(Luke, Leia),
     (Han, Leia),
     (Leia, Han)}
```

Directed vs. undirected graphs

When the edges in a graph have no direction, the graph is called *undirected*

ected graph.

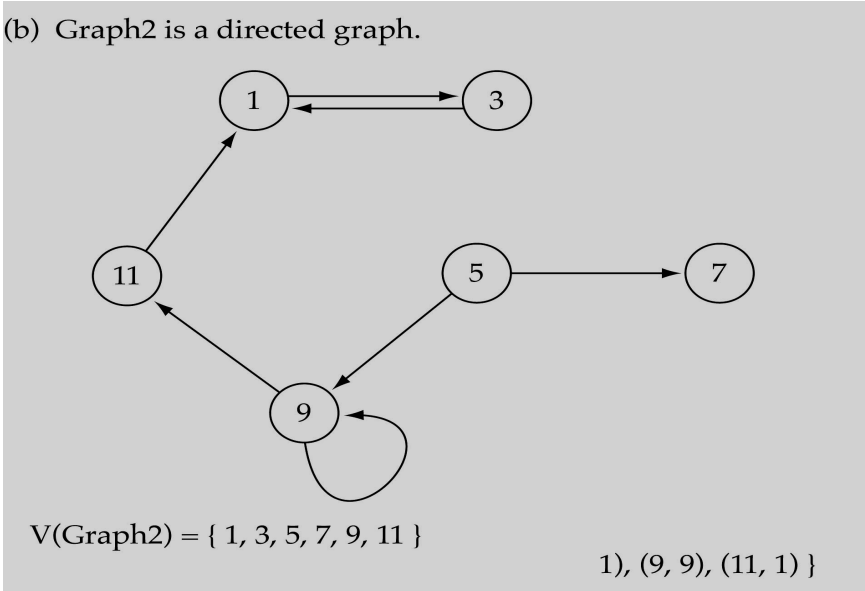


V(Graph1) = { A, B, C, D }
E(Graph1) = { (A, B), (A, D), (B, C), (B, D) }

Directed vs. undirected graphs (cont.)

When the edges in a graph have a direction, the graph is called directed (or digraph).

If the graph is directed, the order of the vertices in each edge is important !!



Graph Representations

Adjacency Matrix
Adjacency Lists
Adjacency Multi-lists

Adjacency Matrix

Let G=(V,E) be a graph with n vertices.
The adjacency matrix of G is a two-dimensional n by n array, say adj_mat

If the edge (vi, vj) is in E(G), adj_mat[i][j]=1

If there is no such edge in E(G), adj_mat[i][j]=0

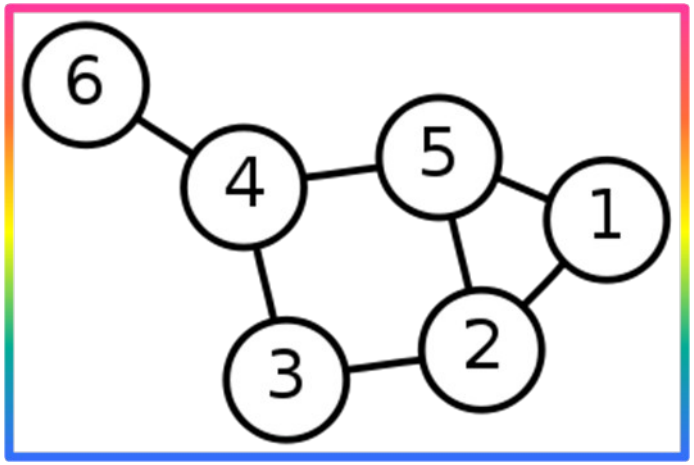
The adjacency matrix for an undirected graph is symmetric; the adjacency matrix for a digraph need not be symmetric

Examples for Adjacency Matrix

A labeled simple graph:

Vertex set V = {1, 2, 3, 4, 5, 6}

Edge set E = {{1,2}, {1,5}, {2,3}, {2,5}, {3,4}, {4,5}, {4,6}}.



	1	**2**	**3**	**4**	**5**	**6**
1	0	***1***	0	0	***1***	0
2	***1***	0	***1***	0	***1***	0
3	0	***1***	0	***1***	0	0
4	0	0	***1***	0	***1***	***1***
5	***1***	***1***	0	***1***	0	0
6	0	0	0	***1***	0	0

Adjacency List

A **|V|**-ary list (array) in which each entry stores a list(linked list) of all adjacent vertices.

Adjacency Multi list

An edge in an undirected graph is represented by two nodes in adjacency list representation.

Adjacency Multi lists lists in which nodes may be shared among several lists.

Graph Traversal

Depth First search:

Pre/Post/In – order traversals are examples of depth-first search.

Nodes are visited deeply on the left-most branches before any nodes are visited on the right-most branches

Visiting the right branches deeply before the left would still be depth-first! Crucial idea is “go deep first!”

Difference in pre/post/in-order is how some computation (*e.g.* printing) is done at current node relative to the recursive calls

In DFS the nodes “being worked on” are kept on a stack

Breadth-First Search:

Level-order traversal is an example of Breadth-First Search
BFS characteristics:
Nodes being worked on maintained in a FIFO Queue, not a stack.
Iterative style procedures often easier to design than recursive procedures
Put root in a Queue
Repeat until Queue is empty:
Dequeue a node
Process it
Add it’s children to queue

Assignment

1)Define Adjacency matrix?
2)What do you mean by spanning trees?
3)How do you represent the binary tree in the computer's memory?
4)Explain how to find shortest path with the help of Dijikstra algorithm?
5)Explain the concept of graph traversal? Write an algorithm for its types?
6)How will you calculate the minimum cost of the spanning tree using kruskal algorithm?
7)Explain the prim's algorithm with examples?
8)Explain the warshal algorithm with examples?

Tutorials

1. For an undirected graph with n vertices and e edges, what is the sum of the degree of each vertex?
2. What do you understand by Transitive closure of graph?
3. Which data structure is used to implement the BFS and DFS? Explain.
4. Find single source shortest path using Dijkastra's algorithm with suitable example.
5. What is minimum spanning tree? Give kruskal algorithm to find a minimum cost spanning tree with suitable example.

Outcome

To analyze algorithms and to determine algorithm correctness and time efficiency.

Understand various terminologies and traversals of graphs and use them for various applications.