# MsDS - Data Structures and Algorithms

Module No: 6

## Revision from Last Lecture

1. HashTables are efficient data structures that use distinct keys to establish a correspondence with values, while HashMaps allow the inclusion of null values and keys.
2. Binary Search Trees (BSTs) guarantee the ordering of nodes, with all nodes in the left subtree having values less than the root, and all nodes in the right subtree having values larger than the root.
3. Priority queues handle elements based on their priority, with efficient access or removal of the element with the highest or lowest priority. Heaps, particularly binary heaps, are commonly used for priority queues.

## Expected Learning Outcomes

1. Understand how Tries are structured and used for effective string manipulation and retrieval.
2. Understand the fundamentals of Huffman Coding and how it is used to compress data using Binary Trees.
3. Understand the value of graphs in describing relationships and investigate different methods of storing them.
4. Understand the fundamental graph operations, such as identifying Paths, performing Traversals, and evaluating Connectivity.

## Tries

* Trie is a tree-based data structure used to efficiently handle a dynamic collection of strings.
* Tries store strings by connecting nodes to letters, allowing for efficient retrieval and existence checks.
* Tries are commonly used for dictionaries and phonebooks, providing auto-completions and faster search operations compared to balanced trees.

## Huffman Coding using Binary Trees

* Huffman Coding is a leading method for achieving lossless compression in data compression.
* Binary trees are crucial in the implementation of Huffman Coding, with nodes representing symbols and their frequencies.
* Huffman Coding is widely used in file compression programs, but it may have limitations when dealing with small datasets or fluctuating symbol frequencies.

## Graphs for Capturing Relations; Storing Graphs

* Graphs are mathematical structures that represent interactions between elements, such as social networks or computer networks.
* There are two common techniques for storing graphs: adjacency matrices and adjacency lists.
* Graphs have practical applications in various fields, including navigation systems and recommendation engines.

## Basic Operations on Graphs

* Connectivity in graphs refers to the existence of routes connecting vertices within a network.
* Graph traversal techniques include Depth-First Search (DFS) and Breadth-First Search (BFS).
* Paths in graphs are ordered series of vertices connected by edges, with the shortest path being the one with the fewest edges or minimal total weight in weighted graphs.

## Important Terminologies

* Tries: Efficient tree-like data structures for string storage and retrieval.
* Huffman Coding: The use of binary trees in a lossless data compression technique.
* Graphs: The structures that depict items and their relationships.
* Connectivity: The metric that quantifies the level of interconnectivity among the various components of a graph.
* Traversals: Methods such as Depth-First Search (DFS) and Breadth-First Search (BFS) are used in a systematic manner to traverse all vertices inside a graph.
* Paths: Routes connecting two vertices in a graph without revisiting any vertex.

## Summary

* Tries are effective for string operations and offer advantages in performance and space optimisation.
* Huffman Coding uses Binary Trees to achieve data compression by assigning shorter codes to frequently occurring characters.
* Graphs are essential for representing interactions and connections, providing efficient storage for relational data. Operations on graphs are foundational for algorithms and real-world applications.

THANK YOU