Name : Ahmed Sayed Kamel
Track : Digital IC Design

Data Structure and Algorithms Report

Summary:

Data structures are organized formats for storing and managing data efficiently. They enable quick access, modification and manipulation of data, optimizing performance for various applications. Common types include arrays, linked lists, trees, hash tables and graphs. This report is focusing on two main data structures: Hash tables and Graphs.

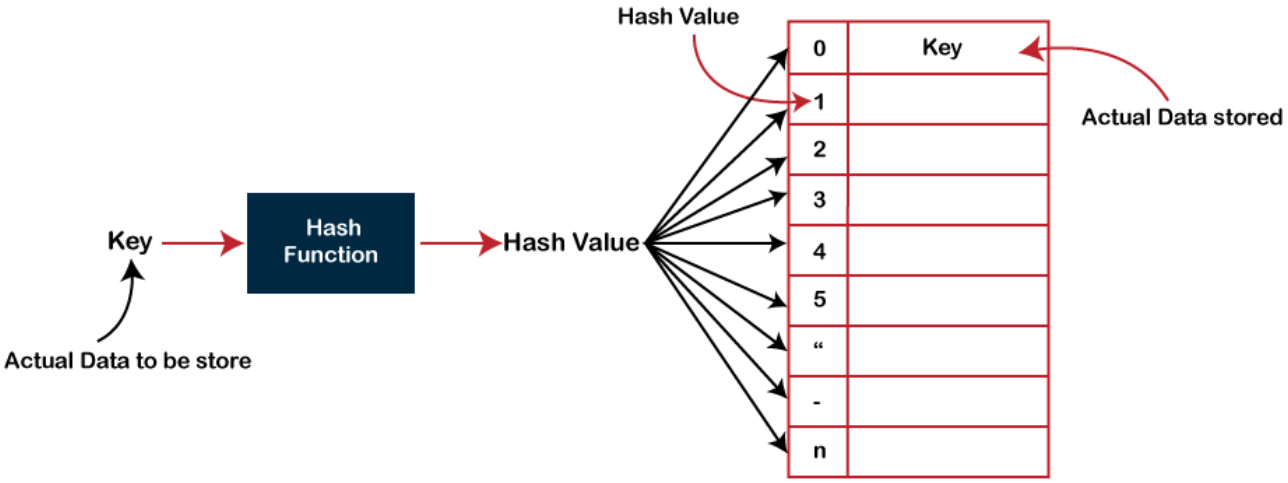
1. Hash Tables

- What is a Hash Table?

A Hash Table is a data structure that allows for efficient insertion, deletion and lookup of elements using a hash function. The key is hashed into an index, where the corresponding value is stored.

Key operations:

- **Insert**: Adding a value by key.
- **Search**: Retrieving a value by key.
- **Delete**: Removing a value by key.



- Figure shows how hash tables works

- Hashing Process

The process of converting a key into an index (hash code) in a hash table to quickly access the corresponding value. It uses a **hash function** to map data to a fixed-size array, minimizing **collisions** and ensuring efficient lookups.

- **Hash Function**: The function takes a key and converts it to an integer index in the hash table array. A good hash function distributes keys evenly across the array.
- **Collisions**: When two keys hash to the same index, a collision occurs. There are two main methods for collision resolution:
    - **Chaining**: Store multiple elements at the same index using a linked list.
    - **Open Addressing**: Find another available slot within the table.

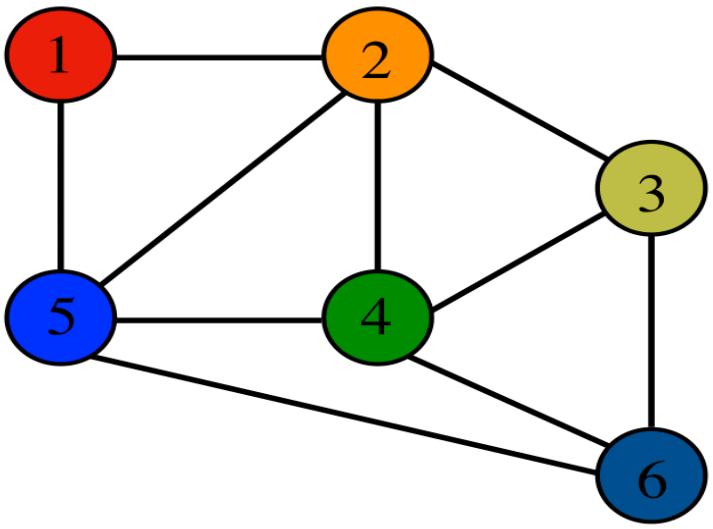
- Applications of Hash Tables

1. **Database Indexing**: Enables fast data retrieval by key.
2. **Caching**: Used in many web services to store frequently accessed data.

2. Graphs

- What is a Graph?

A Graph is a collection of nodes (vertices) and edges (connections between nodes). It can be used to model networks like social media connections, road maps, computer networks, etc.



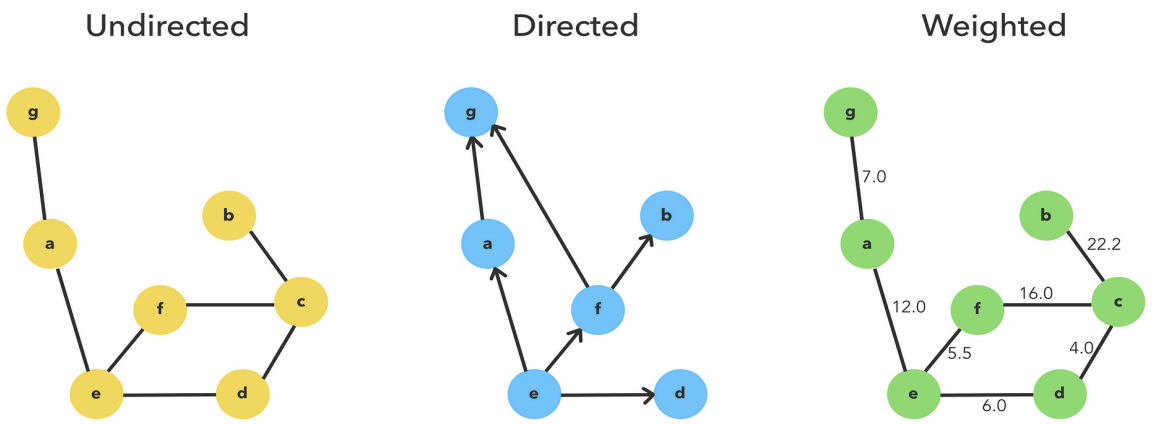
- Figure shows graphs

- Types of Graphs

- **Directed Graph**: In a directed graph, edges have a direction, meaning they go from one vertex to another. For example, in a web page link structure, a directed edge would represent a link from one page to another.

- **Undirected Graph**: In this type of graph, the edges do not have a direction. A connection between two nodes signifies mutual relationships, such as friendship in a social network.

- **Weighted Graph**: Each edge in a weighted graph has an associated weight or cost, representing the cost of traversing from one vertex to another. For example, a road map might use weights to represent distances between cities.



- Figure shows types graphs

- Applications of Graphs

1. **Routing Algorithms**: In networks or GPS navigation systems, graphs represent routes between locations.
2. **Social Networks**: Modeling friendships or connections between individuals.