1. Definition of Graph and Its Types

A graph is a non-linear data structure used to represent relationships between objects. It consists of a finite set of vertices and a set of edges that connect pairs of vertices. Graphs are used to model real-world systems such as networks, maps, and connections. There are different types of graphs. An undirected graph has edges with no direction. A directed graph has edges with a specific direction. A weighted graph has edges associated with weights. An unweighted graph has no weights. A simple graph has no loops or multiple edges. A complete graph has an edge between every pair of vertices.

2. Edges and Vertices

Vertices are the basic units or nodes of a graph. Edges are the connections between the vertices. Each edge represents a relationship between two vertices. For example, in a graph representing cities, the cities are vertices and the roads connecting them are edges. Vertices store data, while edges define how the data elements are related to each other.

3. Degree of a Node

The degree of a node is the number of edges connected to that node. In an undirected graph, the degree is the total number of edges connected to the vertex. In a directed graph, degree is divided into indegree and outdegree. Indegree is the number of edges coming into the node, and outdegree is the number of edges going out from the node. Degree helps in analyzing the importance and connectivity of a node.

4. Difference Between Directed Graph and Weighted Graph

A directed graph is a graph in which edges have a direction, meaning they point from one vertex to another. A weighted graph is a graph in which edges have associated values or costs. Direction and weight represent different concepts. For example, a directed graph can represent one-way roads, while a weighted graph can represent distances or costs between locations.

5. Adjacency List Representation

Adjacency list representation stores a list of adjacent vertices for each vertex in the graph. It is memory efficient for sparse graphs. Below is an example of adjacency list representation.

```
Vertex A: B, C
Vertex B: A, D
Vertex C: A, D
Vertex D: B, C
```

6. Difference Between Spanning Tree and Minimum Spanning Tree

A spanning tree is a subgraph that connects all the vertices of a graph without forming cycles. It contains all vertices with minimum number of edges. A minimum spanning tree is a spanning tree with the minimum possible total edge weight. While every minimum spanning tree is a spanning tree, not all spanning trees are minimum spanning trees.

7. Applications of Spanning Tree

Spanning trees are widely used in computer science. They are used in network design to minimize wiring or connection costs. Spanning trees are used in routing algorithms and broadcast networks. They help in finding optimal paths and eliminating loops in networks. They are also used in clustering and image processing.