**Graph**: (V , E) consists of V , a nonempty set of vertices (or nodes) and E, a set of

edges. Each edge has either one or two vertices associated with it, called its endpoints.

**Infinite Graph**: A graph with an infinite vertex set or an infinite number of edges

**Finite Graph:** A graph with a finite vertex set and a finite edge set

**Simple Graph:** A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices

**Multigraphs:** Graphs that may have multiple edges connecting the same vertices

**Pseudographs:** Graphs that may include loops, and possibly multiple edges connecting the same pair of vertices or a vertex to itself

**Directed Graph**: (or digraph) (V , E) consists of a nonempty set of vertices V and a set of

directed edges (or arcs) E. Each directed edge is associated with an ordered pair of vertices.

The directed edge associated with the ordered pair (u, v) is said to start at u and end at v.

**Simple Directed Graph:** A directed graph with no loops and no multiple directed edges

**Directed Multigraphs:** Directed graphs that may have multiple directed edges from a vertex to a second (possibly the same) vertex are used to model such networks.

**Mixed Graph:**  A graph with both directed and undirected edges

**social networks:** Graphs that are extensively used to model social structures based on dif-

ferent kinds of relationships between people or groups of people.

**Isolated:** A vertex of degree zero

**Pendant**: if and only if a vertex has degree one.

**complete graph**: a simple graph that contains exactly one edge between each pair of distinct vertices.

**Non Complete graph:** A simple graph for which there is at least one

pair of distinct vertex not connected by an edge

**Bipartite:** vertex set V can be partitioned into two disjoint sets V1 and V2 such that every edge in the graph connects a vertex in V1 and a vertex in V2

**Serial :** the algorithms written to solve problems were designed to perform one step at a time

**Parallel processing** : uses computers made up of many separate processors, each

with its own memory, helps overcome the limitations of computers with a single processor.

**Parallel algorithm :** breaks a problem into a number of subproblems that can be solved

concurrently, can then be devised to rapidly solve problems using a computer with multiple

Processors.

**Hops**:  a large number of intermediate links for processors to share information.

**Subgraph:** When edges and vertices are removed from a graph, without removing endpoints of any remaining edges, a smaller graph is obtained.

**edge contraction**: removes an edge e with endpoints u and v and merges u and w into a new single vertex w, and for each edge with u or v as an endpoint replaces the edge with one with w as endpoint in place of u or v and with the same second endpoint.

**union of the graphs:** graph that contains all the vertices and edges of these graphs

**Isomorphic**: two graphs have exactly the same form, in the sense that there is a one-to-one

correspondence between their vertex sets that preserves edges.

**adjacency lists**: specify the vertices that are adjacent to each vertex of the graph.

**graph invariant:** two graphs are not isomorphic if we can find a property only one of the two graphs has, but that is preserved by isomorphism.

**Path**: a sequence of edges that begins at a vertex of a graph and travels from

vertex to vertex along edges of the graph.

**Connected:** An undirected graph’s path between every pair of distinct

vertices of the graph.

**Disconnected:** An undirected graph that is not connected is called. We

say that we disconnect a graph when we remove vertices or edges, or both, to produce a

disconnected subgraph.

**Cut vertices**: the removal from a graph of a vertex and all incident edges produces a subgraph

with more connected components.

**Non separable graphs:** Connected graphs without cut vertices and can be thought of as more

connected than those with a cut vertex.

**Vertex cut:** the vertex connectivity of a non complete graph G, denoted by κ(G), as the minimum number of vertices in a strongly connected graph

**strongly connected:** if there is a path from a to b and from b to a whenever a and b are vertices in the graph.

**weakly connected**: if there is a path between every two vertices in the underlying undirected graph.

**Euler circuit**: a simple circuit containing every edge of G.

**Hamilton path:** A simple path in a graph G that passes through every vertex exactly once

**Gray Code**: a labeling of the arcs of the circle such that adjacent arcs are labeled with bit

strings that differ in exactly one bit.

**Weighted Graphs:** Graphs that have a number assigned to each edge

**approximation algorithm:** A practical approach to the traveling salesperson problem when there are many vertices. These are algorithms that do not necessarily produce the exact solution to the problem but instead are guaranteed to produce a solution that is close to an exact solution.

**Planar:** a graph that is drawn in the plane without any edges crossing (where

a crossing of edges is the intersection of the lines or arcs representing them at a point other

than their common endpoint).

**Regions:** A planar representation of a graph splits the plane

**Elementary subdivision:** If a graph is planar, so will be any graph obtained by removing an edge {u, v} and adding a new vertex w together with edges {u,w} and {w, v}.

**Homeomorphic:** The graphs G1 = (V1, E1) and G2 = (V2, E2) if they

can be obtained from the same graph by a sequence of elementary subdivisions.

**Dual Graph:** Two regions that touch at only one point are not considered adjacent.

C**hromatic Number**: the least number of colors needed for a coloring of this graph.