**TERMS**

**undirected edge:** an edge associated to a set {u, v}, where u and v are vertices

**directed edge:** an edge associated to an ordered pair (u, v), where u and v are vertices

**multiple edges:** distinct edges connecting the same vertices

**multiple directed edges:** distinct directed edges associated with the same ordered pair (u, v), where u and v are vertices

**loop:** an edge connecting a vertex with itself

**undirected graph:** a set of vertices and a set of undirected edges each of which is associated with a set of one or two of these vertices

**simple graph:** an undirected graph with no multiple edges or loops

**multigraph:** an undirected graph that may contain multiple edges but no loops

**pseudograph:** an undirected graph that may contain multiple edges and loops

**directed graph:** a set of vertices together with a set of directed edges each of which is associated with an ordered pair of vertices

**directed multigraph:** a graph with directed edges that may contain multiple directed edges

**simple directed graph:** a directed graph without loops or multiple directed edges

**adjacent:** Two vertices *u* and *v* in an undirected graph *G* are called *adjacent* (or *neighbors*) in *G* if *u*

and *v* are endpoints of an edge *e* of *G*. Such an edge *e* is called *incident with* the vertices *u*

and *v* and *e* are said to *connect u* and *v*.

**incident:** an edge is incident with a vertex if the vertex is an endpoint of that edge

**deg v (degree of the vertex v in an undirected graph):** The *degree of a vertex in an undirected graph* is the number of edges incident with it, except

that a loop at a vertex contributes twice to the degree of that vertex. The degree of the

vertex *v* is denoted by deg*(v)*.

**deg−(v) (the in-degree of the vertex v in a graph with directed edges):** the number of edges with v as their terminalvertex

**deg+ (v) (the out-degree of the vertex v in a graph with directed edges):** the number of edges with v as their initialvertex

**underlying undirected graph of a graph with directed edges:** the undirected graph obtained by ignoring the directionsof the edges

**Kn (complete graph on n vertices):** A **complete graph on *n* vertices**, denoted by *Kn*, is a simple graph

that contains exactly one edge between each pair of distinct vertices. The graphs *Kn*, for

*n* = 1*,* 2*,* 3*,* 4*,* 5*,* 6, are displayed in Figure 3. A simple graph for which there is at least one

pair of distinct vertices not connected by an edge is called **noncomplete**.

**bipartite graph:** a graph with vertex set that can be partitioned into subsets V1 and V2 so that each edge connects a vertex in V1 and a vertex in V2. The pair (V1, V2) is called a **bipartition**

of V.

**Km, n (complete bipartite graph):** the graph with vertex set partitioned into a subset of m elements and a subset of n elements with two vertices connected by an edge if and only

if one is in the first subset and the other is in the second subset

**Cn (cycle of size n), n ≥ 3:** the graph with n vertices v1, v2, . . ., vn and edges {v1, v2}, {v2, v3}, . . ., {vn−1, vn}, {vn, v1}

**Wn (wheel of size n), n ≥ 3:** the graph obtained from Cn by adding a vertex and edges from this vertex to the original vertices in Cn

**Qn (n-cube), n ≥ 1:** the graph that has the 2n bit strings of length n as its vertices and edges connecting every pair of bit string that differ by exactly one bit

**matching in a graph G:** a set of edges such that no two edges have a common endpoint

**complete matching M from V1 to V2:** a matching such that every vertex in V1 is an endpoint of an edge in M

**maximum matching:** a matching containing the most edges among all matchings in a graph

**isolated vertex:** a vertex of degree zero

**pendant vertex:** a vertex of degree one

**regular graph:** a graph where all vertices have the same degree

**subgraph of a graph G = (V, E):** a graph (W, F), where W is a subset of V and F is a subset of E

**G1 ∪G2 (union of G1 and G2):** the graph (V1 ∪ V2, E1 ∪E2), where G1 = (V1, E1) and G2 = (V2, E2)

**adjacency matrix:** a matrix representing a graph using the adjacency of vertices

**incidence matrix:** a matrix representing a graph using the incidence

of edges and vertices

**isomorphic simple graphs:** the simple graphs G1 = (V1, E1) and G2 = (V2, E2) are isomorphic if there exists a one-to-one correspondence f from V1 to V2 such that

{f (v1), f (v2)} ∈ E2 if and only if {v1, v2} ∈ E1 for all v1 and v2 in V1

**invariant for graph isomorphism:** a property that isomorphic graphs either both have, or both do not have

**path from u to v in an undirected graph:** a sequence of edges e1, e2, . . ., en, where ei is associated to {xi, xi+1} for i = 0, 1, . . ., n, where x0 = u and xn+1 = v

**path from u to v in a graph with directed edges:** a sequence of edges e1, e2, . . ., en, where ei is associated to (xi, xi+1) for i = 0, 1, . . ., n, where x0 = u and xn+1 = v

**simple path:** a path that does not contain an edge more than once

**circuit:** a path of length n ≥ 1 that begins and ends at the same vertex

**connected graph:** an undirected graph with the property that there is a path between every pair of vertices

**cut vertex of G:** a vertex v such that G − v is disconnected

**cut edge of G:** an edge e such that G − e is disconnected

**nonseparable graph:** a graph without a cut vertex

**vertex cut of G:** a subset V \_ of the set of vertices of G such that G − V\_ is disconnected

**κ(G) (the vertex connectivity of G):** the size of a smallest vertex cut of G

**k-connected graph**: a graph that has a vertex connectivity no smaller than k

**edge cut of G:** a set of edges E \_ of G such that G − E\_ is disconnected

**λ(G) (the edge connectivity of G):** the size of a smallest edge cut of G

**connected component of a graph G:** a maximal connected subgraph of G

**strongly connected directed graph:** a directed graph with the property that there is a directed path from every vertex to every vertex

**strongly connected component of a directed graph G:** a maximal strongly connected subgraph of G

**Euler path:** a path that contains every edge of a graph exactly once

**Euler circuit:** a circuit that contains every edge of a graph exactly once

**Hamilton path:** a path in a graph that passes through each vertex exactly once

**Hamilton circuit:** a circuit in a graph that passes through each vertex exactly once

**weighted graph:** a graph with numbers assigned to its edges

**shortest-path problem:** the problem of determining the path in a weighted graph such that the sum of the weights of the edges in this path is a minimum over all paths between specified vertices

**traveling salesperson problem:** the problem that asks for the circuit of shortest total length that visits every vertex of a weighted graph exactly once

**planar graph:** a graph that can be drawn in the plane with no crossings

**regions of a representation of a planar graph:** the regions the plane is divided into by the planar representation of the graph

**elementary subdivision:** the removal of an edge {u, v} of an undirected graph and the addition of a new vertex w together with edges {u, w} and {w, v}

**homeomorphic:** two undirected graphs are homeomorphic if they can be obtained from the same graph by a sequence of elementary subdivisions

**graph coloring:** an assignment of colors to the vertices of a graph so that no two adjacent vertices have the same color

**chromatic number:** the minimum number of colors needed in a coloring of a graph

**IMPORTANT:** A simple graph is bipartite if and only if it is possible to assign one of two different colors to

each vertex of the graph so that no two adjacent vertices are assigned the same color.

***Proof:*** First, suppose that= *(V, E)* is a bipartite simple graph. Then *V* = *V*1 ∪ *V*2, where *V*1and *V*2 are disjoint sets and every edge in *E* connects a vertex in *V*1 and a vertex in *V*2. If we assign one color to each vertex in *V*1 and a second color to each vertex in *V*2, then no two adjacent vertices are assigned the same color. Now suppose that it is possible to assign colors to the vertices of the graph using just two colors so that no two adjacent vertices are assigned the same color. Let *V*1 be the set of vertices

assigned one color and *V*2 be the set of vertices assigned the other color. Then, *V*1 and *V*2are disjoint and *V* = *V*1 ∪ *V*2. Furthermore, every edge connects a vertex in *V*1 and a vertex in *V*2 because no two adjacent vertices are either both in *V*1 or both in *V*2. Consequently, *G* is bipartite.

**UNION:** The *union* of two simple graphs *G*1 = *(V*1*, E*1*)* and *G*2 = *(V*2*, E*2*)* is the simple graph with

vertex set *V*1 ∪ *V*2 and edge set *E*1 ∪ *E*2. The union of *G*1 and *G*2 is denoted by *G*1 ∪ *G*2

**UNIT 3:**

**proposition:** a statement that is true or false

**propositional variable:** a variable that represents a proposition

**truth value:** true or false

**￢ *p* (negation of *p*):** the proposition with truth value opposite to the truth value of *p*

**logical operators:** operators used to combine propositions

**compound proposition:** a proposition constructed by combining propositions using logical operators

**truth table:** a table displaying all possible truth values of

propositions

***p* ∨ *q* (disjunction of *p* and *q*):** the proposition “*p* or *q*,” which is true if and only if at least one of *p* and *q* is true

***p* ∧ *q* (conjunction of *p* and *q*):** the proposition “*p* and *q*,” which is true if and only if both *p* and *q* are true

***p* ⊕ *q* (exclusive or of *p* and *q*):** the proposition “*p XOR q*,” which is true when exactly one of *p* and *q* is true

***p* → *q* (*p* implies *q*):** the proposition “if *p*, then *q*,” which is false if and only if *p* is true and *q* is false

**converse of *p*→*q*:** the conditional statement *q* → *p*

**contrapositive of *p*→*q*:** the conditional statement ￢*q* →￢*p*

**inverse of *p*→*q*:** the conditional statement ￢*p* →￢*q*

***p* ↔ *q* (biconditional):** the proposition “*p* if and only if *q*,” which is true if and only if *p* and *q* have the same truth value

**bit:** either a 0 or a 1

**Boolean variable:** a variable that has a value of 0 or 1

**bit operation:** an operation on a bit or bits

**bit string:** a list of bits

**bitwise operations:** operations on bit strings that operate on each bit in one string and the corresponding bit in the other string

**logic gate:** a logic element that performs a logical operation on one or more bits to produce an output bit

**logic circuit:** a switching circuit made up of logic gates that produces one or more output bits

**tautology:** a compound proposition that is always true

**contradiction:** a compound proposition that is always false

**contingency:** a compound proposition that is sometimes true and sometimes false

**consistent compound propositions:** compound propositions for which there is an assignment of truth values to the variables that makes all these propositions true

**satisfiable compound proposition:** a compound proposition for which there is an assignment of truth values to its variables that makes it true

**logically equivalent compound propositions:** compound propositions that always have the same truth values

**predicate:** part of a sentence that attributes a property to the subject

**propositional function:** a statement containing one or more variables that becomes a proposition when each of its variables is assigned a value or is bound by a quantifier

**domain (or universe) of discourse:** the values a variable in a propositional function may take

**∃*x P*(*x*) (existential quantification of *P*(*x*)):** the proposition that is true if and only if there exists an *x* in the domain such that *P(x)* is true

**∀*xP*(*x*) (universal quantification of *P*(*x*)):** the proposition that is true if and only if *P(x)* is true for every *x* in the

domain

**logically equivalent expressions:** expressions that have the same truth value no matter which propositional functions and domains are used

**free variable:** a variable not bound in a propositional function

**bound variable:** a variable that is quantified

**scope of a quantifier:** portion of a statement where the quantifier binds its variable

**argument:** a sequence of statements

**argument form:** a sequence of compound propositions involving propositional variables

**premise:** a statement, in an argument, or argument form, other than the final one

**conclusion:** the final statement in an argument or argument form

**valid argument form:** a sequence of compound propositions involving propositional variables where the truth of all the premises implies the truth of the conclusion

**valid argument:** an argument with a valid argument form

**rule of inference:** a valid argument form that can be used in the demonstration that arguments are valid

**fallacy:** an invalid argument form often used incorrectly as a rule of inference (or sometimes, more generally, an incorrect

argument)