

9 Counting And Probability

9.1 Introduction

9.1.1 Sample Space

A **sample space** is the set of all possible outcomes of a random process or experiment. An **event** is a subset of a sample space.

9.1.2 Equally Likely Probability Formula

If S is a finite sample space in which all outcomes are equally likely and E is an event in S , then the **probability of E**, denoted $\mathbf{P(E)}$, is

$$P(E) = \frac{\text{the number of outcomes in } E}{\text{the total number of outcomes in } S} \quad (1)$$

The notation is as follows: For any finite set A , $N(A)$ denotes the number of elements in A .

9.1.3 The Number of Elements in a List

If m and n are integers and $m \leq n$, then there are $n - m + 1$ integers from m to n inclusive.

9.2 Possibility Trees and the Multiplication Rule

9.2.1 The Multiplication Rule

If an operation consists of k steps and

the first step can be performed n_1 ways, the second step can be performed in n_2 ways [*regardless of how the first step was performed*]; the k th step can be performed in n_k ways [*regardless of how the preceding steps were performed*],

then the entire operation can be performed in $n_1 n_2 \cdots n_k$ ways.

9.2.2 Number of Permutations

For any integer n with $n \geq 1$, the number of permutations of a set with n elements is $n!$.

9.2.3 R-Permutation

An **r-permutation** of a set of n elements is an ordered selection of r elements taken from the set of n elements. The number of r-permutations of a set of n elements is denoted $\mathbf{P(n, r)}$.

If n and r are integers and $1 \leq r \leq n$, then the number of r -permutations of a set of n elements is given by the formula

$$P(n, r) = n(n-1)(n-2) \cdots (n-r+1)$$

or, equivalently,

$$P(n, r) = \frac{n!}{(n-r)!}$$

9.3 Counting Elements of Disjoint Sets: The Addition Rule

The Addition Rule

Suppose a finite set A equals the union of k distinct mutually disjoint subsets A_1, A_2, \dots, A_k . Then:

$$N(A) = N(A_1) + N(A_2) + \cdots + N(A_k).$$

The Difference Rule

If A is a finite set and B is a subset of A , then:

$$N(A - B) = N(A) - N(B).$$

Formula for the probability of the complement of an Event

If S is a finite sample space and A is an event in S , then:

$$P(A^c) = 1 - P(A)$$

The Inclusion/Exclusion Rule for Two or Three Sets

If A , B and C are any finite sets, then

$$N(A \cup B) = N(A) + N(B) - N(A \cap B)$$

and

$$N(A \cup B \cup C) = N(A) + N(B) + N(C) - N(A \cap B) - N(A \cap C) - N(B \cap C) + N(A \cap B \cap C).$$

Pigeonhole Principle

A function from one finite set to a smaller finite set cannot be one-to-one: There must be at least two elements in the domain that have the same image in the co-domain.

Generalized Pigeonhole Principle

For any function f from a finite set X with n elements to a finite set Y with m elements and for any positive integer k , if $k < n/m$, there is some $y \in Y$ such that y is the image of at least $k + 1$ distinct elements of X .

9.3.1 Generalized Pigeonhole Principle (Contrapositive Form)

For any function f from a finite set X with n elements to a finite set Y with m elements and for any positive integer k , if for each $y \in Y$, $f^{-1}(y)$ has at most k elements, then X has at most km elements; in other words, $n \leq km$.

9.3.2 The Pigeonhole Principle

For any function f from a finite set X with n elements to a finite set Y with m elements, if $n > m$, then f is not one-to-one.

9.3.3 One-to-One and Onto for Finite Sets

Let X and Y be finite sets with the same number of elements and suppose f is a function from X to Y . Then f is one-to-one if, and only if, f is onto.

9.5 R-Combination

Let n and r be nonnegative integers with $r \leq n$. An **r-combination** of a set of n elements is a subset of r of the n elements. As indicated in Section 5.1, the symbol

$$\binom{n}{r}$$

which is read n choose r, denotes the number of subsets of size r (r -combinations) that can be chosen from a set of n elements.

9.5.1 Definition

The number of subsets of size r (or r -combinations) that can be chosen from a set of n elements, n , is given by the formula

$$\binom{n}{r} = \frac{P(n, r)}{r!}$$

or, equivalently,

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

where n and r are nonnegative integers with $r \leq n$.

9.5.2 Permutations with sets of Indistinguishable Objects

Suppose a collection consists of n objects of which

n_1 are of type 1 and are indistinguishable from each other. n_2 are of type 2 and are indistinguishable from each other. \vdots n_k are of type k and are indistinguishable from each other.

and suppose that $n_1 + n_2 + \cdots + n_k = n$. Then the number of distinguishable

$$\frac{\binom{n}{n_1} \binom{n-n_1}{n_2} \binom{n-n_1-n_2}{n_3} \cdots \binom{n-n_1-n_2-\cdots-n_{k-1}}{n_k}}{n!} \quad (2)$$

9.6 r-Combinations with Repetition Allowed

An **r-combination with repetition allowed**, or **multiset of size r** , chosen from a set X of n elements is an unordered selection of elements taken from X with repetition allowed. If $X = \{x_1, x_2, \dots, x_n\}$, we write an r -combination with repetition allowed, or multiset of size r , as $[x_{i_1}, x_{i_2}, \dots, x_{i_r}]$, where each x_{i_j} is in X and some of the x_{i_j} may equal.

9.6.1 Theorem

The number of r -combinations with repetition allowed (multisets of size r) that can be selected from a set of n elements is

$$\binom{r+n-1}{r}$$

This equals the number of ways r objects can be selected from n categories of objects with repetition allowed.

9.7 Graphs: Definitions and Basic Properties

9.7.1 Graph Terminology

A **graph** G consists of two finite sets: a nonempty set $V(G)$ of **vertices** and a set $E(G)$ of **edges**, where each edge is associated with a set consisting of either one or two vertices called its **endpoints**. The correspondence from edges to endpoints is called the **edge-endpoint function**.

An edge with just one endpoint is called a **loop**, and two or more distinct edges with the same set of endpoints are said to be **parallel**. An edge is said to **connect** its endpoints; two vertices that are connected by an edge are called **adjacent**; and a vertex that is an endpoint of a loop is said to be **adjacent to itself**.

An edge is said to be **incident on** each of its endpoints, and two edges incident on the same endpoint are called **adjacent**. A vertex on which no edges are incident is called **isolated**.

9.7.2 Directed Graph / Digraph

A **directed graph**, or **digraph**, consists of two finite sets: a nonempty set $V(G)$ of vertices and a set $D(G)$ of directed edges, where each is associated with an ordered pair of vertices called its **endpoints**. If edge e is associated with the pair (v, w) of vertices, then e is said to be the **(directed) edge** from v to w .

9.7.3 Simple Graph

A **simple graph** is a graph that does not have any loops or parallel edges. In a simple graph, an edge with endpoints v and w is denoted v, w .

9.7.4 Complete Graph of n Vertices

Let n be a positive integer. A **complete graph on n vertices**, denoted K_n , is a simple graph with n vertices and exactly one edge connecting each pair of distinct vertices.

9.7.5 Complete Bipartite Graph on (m, n) Vertices

Let m and n be positive integers. A complete bipartite graph on (m, n) vertices, denoted $K_{m,n}$, is a simple graph with distinct vertices v_1, v_2, \dots, v_m and w_1, w_2, \dots, w_n that satisfies the following properties: For all $i, k = 1, 2, \dots, m$ and for all $j, l = 1, 2, \dots, n$,

1. There is an edge from each vertex v_i to each vertex w_j .
2. There is no edge from any vertex v_i to any other vertex v_k
3. There is no edge from any vertex w_j to any other vertex w_l

9.7.6 Subgraph

A graph H is said to be a **subgraph** of a graph G if, and only if, every vertex in H is also a vertex in G , every edge in H is also an edge in G , and every edge in H has the same endpoints as it has in G .

9.7.7 Degree

Let G be a graph and v a vertex of G . The **degree of v** , denoted $\deg(v)$, equals the number of edges that are incident on v , with an edge that is a loop counted twice. The **total degree of G** is the sum of the degrees of all the vertices of G .

9.7.8 The Handshake Theorem

If G is any graph, then the sum of the degrees of all the vertices of G equals twice the number of edges of G . Specifically, if the vertices of G are v_1, v_2, \dots, v_n , where n is a nonnegative integer, then

$$\text{The total degree of } G = \deg(v_1) + \deg(v_2) + \dots + \deg(v_n) \quad (3)$$

$$= 2 \times (\text{the number of edge of } G) \quad (4)$$

This means that **the total degree of a graph is even.**

9.7.9 Proposition

In any graph there are an even number of vertices of odd degree.

9.7.10 Complement

If G is a simple graph, the **complement of G** , denoted G' , is obtained as follows: The vertex set of G' is identical to the vertex set of G . However, two distinct vertices v and w of G' are connected by an edge if, and only if, v and w are not connected by an edge in G .

9.8 Trails, Paths, and Circuits

9.8.1 Definitions

Let G be a graph, and let v and w be vertices in G .

Walk A **walk from v to w** finite alternating sequence of adjacent vertices and edges of G . Thus a walk has the form

$$v_0 e_1 v_1 e_2 \dots v_{n-1} e_n v_n \quad (5)$$

where the v s represent vertices, the e s represent edges, $v_0 = v, v_n = w$, and for all $i = 1, 2, \dots, n$, v_{i-1} and v_i are the endpoints of e_i . The **trivial walk from v to v** consists of the single vertex v .

Trail A **trail** from v to w is a walk from v to w that does not contain a repeated edge.

Path A **path** from v to w is a trail that does not contain a repeated vertex.

Closed Walk A **closed walk** is a walk that starts and ends at the same vertex.

Circuit A **circuit** is a closed walk that contains at least one edge and does not contain a repeated edge.

Simple Circuit A **simple circuit** is a circuit that does not have any other repeated vertex except the first and last.

	Repeated Edge?	Repeated Vertex?	Starts and Ends at Same Point?	Must Contain At Least One Edge?
Walk	allowed	allowed	allowed	no
Trail	no	allowed	allowed	no
Path	no	no	no	no
Closed Walk	allowed	allowed	yes	no
Circuit	no	allowed	yes	yes
Simple Circuit	no	first and last only	allowed	yes

9.8.2 Connected

Let G be a graph. Two **vertices v and w of G are connected** if, and only if, there is a walk from v to w . The **graph G is connected** if, and only if, given *any* two vertices v and w in G , there is a walk from v to w . Symbolically,

$$G \text{ is connected} \Leftrightarrow \forall \text{ vertices } v, w \in V(G), \exists \text{ a walk from } v \text{ to } w. \quad (6)$$

Let G be a graph.

- If G is connected, then any two distinct vertices of G can be connected by a path.
- . If vertices v and w are part of a circuit in G and one edge is removed from the circuit, then there still exists a trail from v to w in G .
- If G is connected and G contains a circuit, then an edge of the circuit can be removed without disconnecting G .

A graph H is a connected component of a graph G if, and only if,

1. H is subgraph of G ;
2. H is connected; and
3. no connected subgraph of G has H as a subgraph and contains vertices or edges that are not in H .

9.8.3 Euler Circuit

Let G be a graph. An **Euler circuit** for G is a circuit that contains every vertex and every edge of G . That is, an Euler circuit for G is a sequence of adjacent vertices and edges in G that has at least one edge, starts and ends at the same vertex, uses every vertex of G at least once, and uses every edge of G exactly once.

If a graph has an Euler circuit, then every vertex of the graph has positive even degree. If some vertex of a graph has odd degree, then the graph does not have an Euler circuit.

9.8.4 Euler Algorithm

If a graph G is connected and the degree of every vertex of G is a positive even integer, then G has an Euler circuit.

Proof: Suppose that G is any connected graph and suppose that every vertex of G is a positive even integer. *[We must find an Euler circuit for G .]* Construct a circuit C by the following algorithm:

Step 1: Pick any vertex v of G at which to start. *[This step can be accomplished because the vertex set of G is nonempty by assumption.]*

Step 2: Pick any sequence of adjacent vertices and edges, starting and ending at v and never repeating an edge. Call the resulting circuit C . *[This step can be performed for the following reasons: Since the degree of each vertex of G is a positive even integer, as each vertex of G is entered by traveling on one edge, either the vertex is v itself and there is no other unused edge adjacent to v , or the vertex can be exited by traveling on another previously unused edge. Since the number of edges of the graph is finite (by definition of graph), the sequence of distinct edges cannot go on forever. The sequence can eventually return to v because the degree of v is a positive even integer, and so if an edge connects v to another vertex, there must be a different edge that connects back to v .]*

Step 3: Check whether C contains every edge and vertex of G . If so, C is an Euler circuit, and we are finished. If not, perform the following steps.

Step 3a: Remove all edges of C from G and also any vertices that become isolated when the edges of C are removed. Call the resulting subgraph G . *[Note that G may not be connected (as illustrated in Figure 10.2.4), but every vertex of G has positive, even degree (since removing the edges of C removes an even number of edges from each vertex, the difference of two even integers is even, and isolated vertices with degree 0 were removed.)]*

Step 3b: Pick any vertex w common to both C and G . *[There must be at least one such vertex since G is connected. (See exercise 44.) (In Figure 10.2.4 there are two such vertices: u and w .)]*

Step 3c: Pick any sequence of adjacent vertices and edges of G , starting and ending at w and never repeating an edge. Call the resulting circuit C . *[This can be done since each vertex of G has positive, even degree and G is finite. See the justification for step 2.]*

Step 3d: Patch C and C together to create a new circuit C as follows: Start at v and follow C all the way to w . Then follow C all the way

back to w . After that, continue along the untraveled portion of C to return to v . [The effect of executing steps 3c and 3d for the graph of Figure 10.2.4 is shown in Figure 10.2.5.]

Step 3e: Let $C = C$ and go back to step 3.

Since the graph G is finite, execution of the steps outlined in this algorithm must eventually terminate. At that point an Euler circuit for G will have been constructed. (Note that because of the element of choice in steps 1, 2, 3b, and 3c, a variety of different Euler circuits can be produced by using this algorithm.)

9.8.5 Theorem

A graph G has an Euler circuit if, and only if, G is connected and every vertex of G has positive even degree.

9.8.6 Corollary

Let G be a graph, and let v and w be two distinct vertices of G . There is an Euler path from v to w if, and only if, G is connected, v and w have odd degree, and all other vertices of G have positive even degree.

9.8.7 Hamiltonian Circuit

Given a graph G , a Hamiltonian circuit for G is a simple circuit that includes every vertex of G . That is, a Hamiltonian circuit for G is a sequence of adjacent vertices and distinct edges in which every vertex of G appears exactly once, except for the first and the last, which are the same.

If a graph G has a Hamiltonian circuit, then G has a subgraph H with the following properties:

1. H contains every vertex of G .
2. H is connected.
3. H has the same number of edges as vertices.
4. Every vertex of H has degree 2.

9.9 Matrix Representations of Graphs

An $\mathbf{m} \times \mathbf{n}$ (read m by n) **matrix \mathbf{A} over a set \mathbf{S}** is a rectangular array of elements of \mathbf{S} arranged into m rows and n columns.

We write $\mathbf{A} = (a_{ij})$

9.9.1 Adjacency Matrix

Let G be a directed graph with ordered vertices v_1, v_2, \dots, v_n . The adjacency matrix of G is the $n \times n$ matrix $\mathbf{A} = (a_{ij})$ over the set of nonnegative integers such that

$$a_{ij} = \text{the number of arrows from } v_i \text{ to } v_j \text{ for all } i, j = 1, 2, \dots, n. \quad (7)$$

9.9.2 Adjacency Matrix

Let G be an undirected graph with ordered vertices v_1, v_2, \dots, v_n . The adjacency matrix of G is the $n \times n$ matrix $A = (a_{ij})$ over the set of nonnegative integers such that

$$\begin{aligned} a_{ij} &= \text{the number of edges connecting } v_i \text{ and } v_j \\ &\text{for all } i, j = 1, 2, \dots, n. \end{aligned} \quad (8)$$

9.9.3 Symmetric

An $n \times n$ square matrix $A = (a_{ij})$ is called **symmetric** if, and only if, for all $i, j = 1, 2, \dots, n$,

$$a_{ij} = a_{ji} \quad (9)$$

9.9.4 Identity Matrix

For each positive integer n , the $n \times n$ **identity matrix**, denoted $I_n = (\delta_{ij})$ or just \mathbf{I} (if the size of the matrix is obvious from context), is the $n \times n$ matrix in which all the entries in the main diagonal are 1s and all other entries are 0s. In other words,

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad \text{for all } i, j = 1, 2, \dots, n$$

9.9.5 The Number of Walks

If G is a graph with vertices v_1, v_2, \dots, v_m and A is the adjacency matrix of G , then for each positive integer n and for all integers $i, j = 1, 2, \dots, m$,

the ij th entry of A^n = the number of walks of length n from v_i to v_j .

9.5 Trees

9.5.1 Terminology

A graph is said to be **circuit-free** if, and only if, it has no circuits. A graph is called a **tree** if, and only if, it is circuit-free and connected. A **trivial tree** is a graph that consists of a single vertex. A graph is called a **forest** if, and only if, it is circuit-free and not connected.

9.5.2 Vertex-Degree Relation

Any tree that has more than one vertex has at least one vertex of degree 1.

9.5.3 Terminal Vertex

Let T be a tree. If T has only one or two vertices, then each is called a **terminal vertex**. If T has at least three vertices, then a vertex of degree 1 in T is called a **terminal vertex** (or a **leaf**), and a vertex of degree greater than 1 in T is called an **internal vertex** (or a **branch vertex**).

9.5.4 Edges And Vertices

For any positive integer n , any tree with n vertices has $n - 1$ edges.

9.6 Connected, Circuit

If G is any connected graph, C is any circuit in G , and any one of the edges of C is removed from G , then the graph that remains is connected.

10.6 Rooted Trees

A **rooted tree** is a tree in which there is one vertex that is distinguished from the others and is called the **root**. The **level** of a vertex is the number of edges along the unique path between it and the root. The **height** of a rooted tree is the maximum level of any vertex of the tree. Given the root or any internal vertex v of a rooted tree, the **children** of v are all those vertices that are adjacent to v and are one level farther away from the root than v . If w is a child of v , then v is called the **parent** of w , and two distinct vertices that are both children of the same parent are called **siblings**. Given two distinct vertices v and w , if v lies on the unique path between w and the root, then v is an **ancestor** of w and w is a **descendant** of v .

10.7 Binary Tree

A **binary tree** is a rooted tree in which every parent has at most two children. Each child in a binary tree is designated either a **left child** or a **right child** (but not both), and every parent has at most one left child and one right child. A **full binary tree** is a binary tree in which each parent has exactly two children.

Given any parent v in a binary tree T , if v has a left child, then the **left subtree** of v is the binary tree whose root is the left child of v , whose vertices consist of the left child of v and all its descendants, and whose edges consist of all those edges of T that connect the vertices of the left subtree. The **right subtree** of v is defined analogously.

10.7.1 Theorem

If k is a positive integer and T is a full binary tree with k internal vertices, then T has a total of $2k + 1$ vertices and has $k + 1$ terminal vertices.

10.7.2 Height And Terminal Vertices

For all integers $h \geq 0$, if T is any binary tree with of height h and t terminal vertices, then

$$t \leq 2^h. \quad (10)$$

Equivalently,

$$\log_2 t \leq h \quad (11)$$

10.8 Spanning Trees and Shortest Paths

A **spanning tree** for a graph G is a subgraph of G that contains every vertex of G and is a tree.

10.8.1 Proposition

- Every connected graph has a spanning tree.
- Any two spanning trees for a graph have the same number of edges.

10.8.2 Terminology

A **weighted graph** is a graph for which each edge has an associated positive real number **weight**. The sum of the weights of all the edges is the **total weight** of the graph. A **minimum spanning tree** for a connected weighted graph is a spanning tree that has the least possible total weight compared to all other spanning trees for the graph. If G is a weighed graph and e is an edge of G , then $\mathbf{w}(e)$ denotes the weight of e and $\mathbf{w}(G)$ denotes the total weight of G .

10.8.3 Kruskals Algorithm

Iteration Number	Edge Considered	Weight	Action Take
------------------	-----------------	--------	-------------