

Discrete Structures 2

Lecture 4:

Counting Theory

Part 1

Dr Abdullah Alharbi

Second Semester
(1447 AH – 2026 AD)

Lecture's content is taken from "Discrete Mathematics and Its Applications" 8th edition by Kenneth H. Rosen. McGraw-Hill, 2019.

Lectures Contents:

- **Counting Theory:**
  - **The Basics of Counting**
  - **The Pigeonhole Principle**
  - Permutations & Combinations
  - Binomial Coefficients & Identities
  - Generalized Permutations & Combinations
  - Generating Permutations & Combinations

Part 1: Basics of Counting & Pigeonhole Principle

- **The Basics of Counting:**
  - Introduction
  - The Product Rule
  - The Sum Rule
  - The Subtraction Rule
  - The Division Rule
  - Tree Diagrams
- **The Pigeonhole Principle:**
  - Introduction
  - The Generalized Pigeonhole Principle

Introduction

- **Counting** theory (aka **Combinatorics**) is an important part of discrete mathematics that studies the arrangements of objects.
- We must **count objects** to solve many different types of problems.
- For example, we must count to:
  - determine the **complexity of algorithms**.
  - determine whether there are *enough* **telephone numbers** or **Internet protocol addresses** to meet demand.
  - determine **DNA sequences**.
  - compute **probabilities of events**.
  - determine the **allowable passwords** on a computer system.
- The **counting of objects with certain properties** (*Enumeration*), is an important part of combinatorics.

The Basics of Counting: Introduction

- We can phrase many **counting problems** in terms of **ordered** or **unordered** ***arrangements of the objects of a set*** with or without **repetitions**.
- These arrangements, called **permutations** and **combinations**, are used in many counting problems.
- **Example:**
  - Suppose that a **password** on a **computer system** consists of six, seven, or eight characters.
  - Each of these characters must be a digit or a letter of the alphabet. Each password must contain at least one digit.
  - ❖ **How many such passwords are there?**
- The techniques needed to answer this question and a wide variety of other **counting problems** will be introduced in this lecture.

Basic Counting Principles: The Product Rule

- **Counting problems** may be:
  - very hard.
  - not obvious.
- We first present two **basic counting principles**:
  - **The Product Rule**.
  - **The Sum Rule**.
- Then we will show how they can be used to solve many different counting problems.
- The **product rule** applies when a procedure is made up of separate tasks.

THE PRODUCT RULE Suppose that a procedure can be broken down into a sequence of two tasks. If there are n_1 ways to do the first task and for each of these ways of doing the first task, there are n_2 ways to do the second task, then there are $n_1 n_2$ ways to do the procedure.

The Product Rule: Examples

- A **count** can be broken down into a sequence of dependent counts:
  - "**each element in the first count is associated with all elements of the second count**".
- **Example:**
  - A new company with just two employees, Sanchez and Patel, rents a floor of a building with **12** offices.
  - How many ways are there to assign different offices to these two employees?
- **Solution:**
  - The procedure of assigning offices to these two employees consists of **(1)** assigning an office to Sanchez, which can be done in **12 ways**, then **(2)** assigning an office to Patel different from the office assigned to Sanchez, which can be done in **11 ways**.
  - By the **product rule**, there are **$12 \cdot 11 = 132$ ways** to assign offices to these two employees.

The Product Rule: Examples

- **Example:**
  - The chairs of an auditorium are to be labelled with an uppercase English letter followed by a positive integer not exceeding 100.
  - What is the largest number of chairs that can be labelled differently?
- **Solution:**
  - The procedure of labelling a chair consists of **two tasks**, namely, **(1)** assigning to the seat one of the 26 uppercase English letters, and then **(2)** assigning to it one of the 100 possible integers.
  - The **product rule** shows that there are **$26 \cdot 100 = 2600$** different ways that a chair can be labelled.
  - Therefore, the largest number of chairs that can be labelled differently is **2600**.

The Product Rule: Examples

- **Example:**
  - There are **32 computers** in a **data center** in the **cloud**.
  - Each of these computers has **24 ports**.
  - How many different computer ports are there in this data center?
- **Solution:**
  - The procedure of choosing a port consists of **two tasks**, first **(1)** picking a computer and then **(2)** picking a port on this computer.
  - Because there are **32 ways** to choose the computer and **24 ways** to choose the port *no matter which computer has been selected*, the **product rule** shows that there are $32 \cdot 24 = 768$ **ports**.

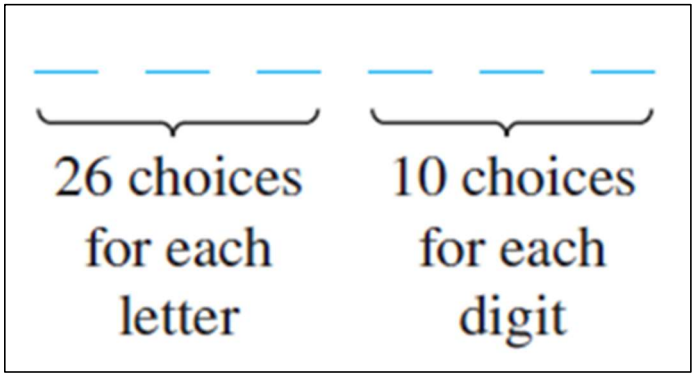
The Product Rule: Examples

- **Example:**
  - How many different **bit strings** of **length seven** are there?
- **Solution:**
  - Each of the **seven bits** can be chosen in **two ways**, because each bit is either 0 or 1.
  - Therefore, the **product rule** shows there are a total of **(2 * 2 * 2 * 2 * 2 * 2 * 2) = 2^7 = 128** different **bit strings of length seven**.

The Product Rule: Examples

Example:

- How many different **license plates** can be made if each plate contains a **sequence** of **three uppercase English letters** followed by **three digits**.
  - *No sequences of letters are prohibited, even if they are obscene.*



Solution:

- There are (1) **26 choices** for each of the three uppercase English letters and (2) **10 choices** for each of the three digits.
- Hence, by the **product rule** there are a total of **$26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 = 17{,}576{,}000$** possible **license plates**.

The Product Rule: Examples

- **Example (Counting Functions):**
  - How many **functions** are there from a **set** with ***m*** elements to a **set** with ***n*** elements?
- **Solution:**
  - A **function** corresponds to a (1) **choice of one of the *n* elements in the codomain** for (2) **each of the *m* elements in the domain**.
  - Hence, by the **product rule** there are $n \cdot n \cdot \cdots \cdot n = n^m$ **functions** from a **set** with ***m*** elements to one with ***n*** elements.
  - For example,
    - There are $5^3 = 125$ different **functions** from a set with **three elements** to a set with **five elements**.

The Product Rule: Examples

- **Example (Counting One-to-One Functions):**
  - How many **one-to-one functions** are there from a **set** with ***m*** elements to one with ***n*** elements?
- **Solution:**
  - First note that when ***m*** **>** ***n*** there are **no one-to-one functions** from a set with ***m*** elements to a set with ***n*** elements.
  - Now let ***m*** ≤ ***n***
  - Suppose the elements in the **domain** are $a_1, a_2, \ldots, a_m$
  - There are **n** ways to choose the value of the function at a_1
  - **Because** the function is **one-to-one**, the value of the function at a_2 can be picked in **(n – 1)** ways (*because the value used for a_1 cannot be used again*).
  - In general, the value of the function at a_k can be chosen in **(n – k + 1)** ways.
  - By the **product rule**, there are $n(n-1)(n-2)\cdots(n-m+1)$ **one-to-one functions** from a set with ***m*** elements to one with ***n*** elements.
  - For example, there are **5 · 4 · 3 = 60 one-to-one functions** from a set with **three elements** to a set with **five elements**.

Product Rule in Terms of **Sets**

- The **product rule** is often phrased in terms of **sets** in this way:
  - If $A_1, A_2, \ldots, A_m$ are **finite sets**, then the *number of elements in the **Cartesian product** of these sets is the product of the number of elements in each set*.
- To relate this to the **product rule**, note that the task of choosing an element in the **Cartesian product** $A_1 \times A_2 \times \cdots \times A_m$ is done by **choosing an element in A_1, an element in A_2 ,…, and an element in A_m**
- By the **product rule** it follows that:

$$|A_1 \times A_2 \times \cdots \times A_m| = |A_1| \cdot |A_2| \cdot \cdots \cdot |A_m|.$$

Basic Counting Principles: The Sum Rule

- A **count** decomposes into a **set of independent counts**
  - "**elements of counts** are **alternatives**", they do not depend on each other.

THE SUM RULE If a task can be done either in one of n_1 ways or in one of n_2 ways, where none of the set of n_1 ways is the same as any of the set of n_2 ways, then there are $n_1 + n_2$ ways to do the task.

- The next two examples illustrate how **the sum rule** is used.

The Sum Rule: Examples

- **Example:**
  - Suppose that either a member of the mathematics **faculty** **or** a **student** who is a mathematics major is chosen as a representative to a university committee.
  - How many different choices are there for this **representative** if there are **37** members of the mathematics **faculty** and **83** mathematics **majors** and no one is both a faculty member and a student?
- **Solution:**
  - There are **37** ways to choose a member of the mathematics **faculty** and there are **83** ways to choose a **student** who is a mathematics major.
    - *Choosing a member of the mathematics faculty is never the same as choosing a student who is a mathematics major* **because no one is both a faculty member and a student**.
  - By **the sum rule** it follows that there are **37 + 83 = 120** **possible ways** to pick this **representative**.

The Sum Rule: Examples

- **Example:**
  - A **student** can choose a **computer project** from one of **three lists**.
  - The **three lists** contain **23**, **15**, and **19** *possible* **projects**, respectively.
    - **No project is on more than one list**.
  - How many possible projects are there to choose from?
- **Solution:**
  - The student can choose a project by selecting a project from:
    - the **first** list,
    - the **second** list, or
    - the **third** list.
  - *Because* no project is on more than one list, by **the sum rule** there are **23 + 15 + 19 = 57** ways to choose a **project**.

The Sum Rule in terms of sets

- The **sum rule** can be phrased in terms of **sets** as:
  - If $\mathbf{A_1}$, $\mathbf{A_2}$,…, $\mathbf{A_m}$ are **pairwise disjoint finite sets**, then **the number of elements in the union of these sets** is the sum of the numbers of elements in the sets.
- To relate this to our statement of **the sum rule**, note there are $|\mathbf{A_i}|$ ways to choose an element from $\mathbf{A_i}$ for ***i = 1, 2, … , m***
- **Because** the sets are pairwise disjoint, when we select an element from one of the sets $\mathbf{A_i}$, we do not also select an element from a different set $\mathbf{A_j}$
- Consequently, by **the sum rule**, because we cannot select an element from two of these sets at the same time, the number of ways to choose an element from one of the sets, which is the number of elements in the **union**, is

$$|A_1 \cup A_2 \cup \cdots \cup A_m| = |A_1| + |A_2| + \cdots + |A_m| \text{ when } A_i \cap A_j = \text{ for all } i, j.$$

More Complex Counting Problems

- Many **counting** problems **cannot be solved** using just **the sum rule** or just **the product rule**.
  - More complex counting problems typically require a **combination of the sum and product rules**.
- **Example:**
  - In a version of the computer language **BASIC**, the **name of a variable** is a **string** of **one or two alphanumeric characters**, where uppercase and lowercase letters are not distinguished.
    - An alphanumeric character is **either** one of the **26** English letters **or** one of the **10** digits.
  - Moreover, a variable name must begin with a letter and must be different from the **five** strings of two characters that are reserved for programming use.
  - How many different variable names are there in this version of BASIC?

More Complex Counting Problems: Examples

- **Solution:**
  - Let **V** equal the number of different variable names in this version of **BASIC**.
  - Let V_1 be the number of these that are one character long and V_2 be the number of these that are two characters long.
  - Then by **the sum rule**, $V = V_1 + V_2$
  - **Note** that $V_1 = 26$
    - Because a one-character variable name must be a **letter**.
  - Furthermore, by the **product rule** there are $(26 \cdot 36)$ **strings** of length two that begin with a letter and end with an alphanumeric character.
  - However, **five** of these are excluded, so $V_2 = 26 \cdot 36 - 5 = 931$
  - Hence, there are $V = V_1 + V_2 = 26 + 931 = 957$ different names for variables in this version of **BASIC**.

More Complex Counting Problems: Examples

- **Example:**
  - Each **user** on a computer system has a **password**, which is six to eight characters long, where each character is an uppercase letter **or** a digit.
  - Each **password** must *contain* at least **one digit**.
  - How many possible passwords are there?

- **Solution:**
  - Let **P** be the total number of possible passwords, and let P_6, P_7, and P_8 denote the number of possible passwords of length **6**, **7**, and **8**, respectively.
  - By **the sum rule**, $P = P_6 + P_7 + P_8$
  - We will now find P_6, P_7, and P_8
  - Finding P_6 directly is difficult.
    - To find P_6 it is easier to find the number of strings of uppercase letters and digits that are six characters long, including those with no digits, and subtract from this the number of strings with no digits.
  - By **the product rule**, the number of strings of six characters is 36^6, and the number of strings with no digits is 26^6
  - Hence, $P_6 = 36^6 - 26^6 = 2{,}176{,}782{,}336 - 308{,}915{,}776 = 1{,}867{,}866{,}560$
  - Similarly, we have $P_7 = 36^7 - 26^7 = 78{,}364{,}164{,}096 - 8{,}031{,}810{,}176 = 70{,}332{,}353{,}920$
  - And, $P_8 = 36^8 - 26^8 = 2{,}821{,}109{,}907{,}456 - 208{,}827{,}064{,}576 = 2{,}612{,}282{,}842{,}880$
    Consequently, $P = P_6 + P_7 + P_8 = 2{,}684{,}483{,}063{,}360$

The Subtraction Rule

- Suppose that a task can be done in one of two ways, but some of the ways to do it are **common to both ways**.
  - In this situation, we **cannot** use **the sum rule** to count the number of ways to do the task.
  - If we add the number of ways to do the tasks in these two ways, we get an **overcount** of the total number of ways to do it, because the ways to do the task that are common to the two ways are counted twice.
  - **Overcounting** is perhaps the most common **enumeration error**.
- To **correctly** count the number of ways to do the two tasks, **we must subtract the number of ways that are counted twice**.
- This leads us to an important counting rule.

THE SUBTRACTION RULE If a task can be done in either n_1 ways or n_2 ways, then the number of ways to do the task is $n_1 + n_2$ minus the number of ways to do the task that are common to the two different ways.

The Subtraction Rule (Inclusion–Exclusion for Two Sets)

- **The subtraction rule** is also known as the **principle of inclusion–exclusion**, especially when it is used to count the number of elements in the **union** of two sets.
- Suppose that A_1 and A_2 are **sets**.
  - Then, there are $|A_1|$ ways to select an element from A_1 and $|A_2|$ ways to select an element from A_2

Inclusion-Exclusion Principle

- The number of ways to select an element from A_1 or from A_2, that is:
  - the number of ways to select an element from their **union**, is:
  - (1) the sum of the number of ways to select an element from A_1, and
  - (2) the number of ways to select an element from A_2,
  - (3) minus the number of ways to select an element that is in both A_1 and A_2
- Because there are $|A_1 \cup A_2|$ ways to select an element in either A_1 or in A_2, and $|A_1 \cap A_2|$ ways to select an element common to both sets, we have

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

Inclusion-Exclusion Principle: Example 1

- **Example 1:**
  - How many bit strings of length **eight** *either* **start** with a **1 bit** or **end** with the **two bits 00**?
- **Solution:**
  - **Figure 2** illustrates the three counting problems we need to solve **before** we can *apply* the **principle of inclusion–exclusion**.
  - We can construct a **bit string of length eight** that either
  - Starts with a 1 bit **or** ends with the two bits 00

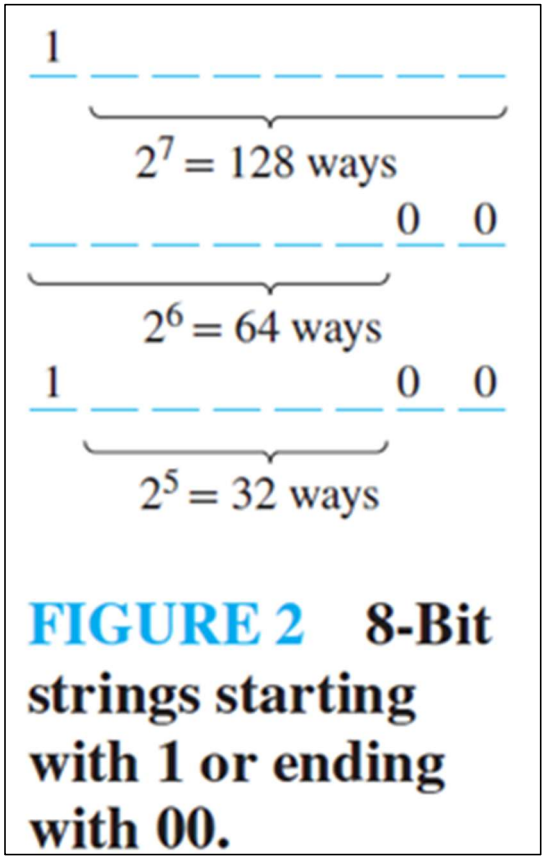


FIGURE 2 **8-Bit strings starting with 1 or ending with 00.**

Inclusion-Exclusion Principle: Example 1 (*cont.*)

- We can *construct* a bit string of length eight that begins with a 1 in $2^7 = 128$ ways.
  - This follows by **the product rule**
    - Because the **first bit** can be chosen in only one way and each of the other **seven bits** can be chosen in two ways.
- Similarly, we can *construct* a bit string of length eight ending with the two bits 00, in $2^6 = 64$ ways.
  - This follows by **the product rule**
    - Because each of the **first six bits** can be chosen in two ways and the **last two bits** can be chosen in only one way.

Inclusion-Exclusion Principle: Example 1 (*cont.*)

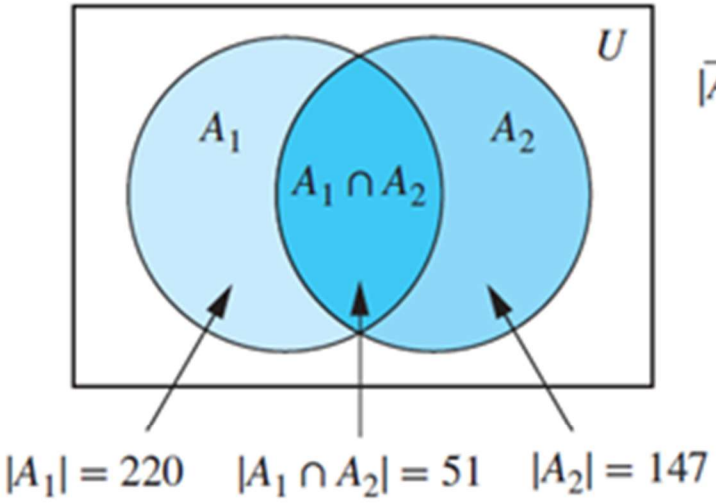
- Some of the ways to *construct* a bit string of length eight starting with a 1 are the **same as the ways** to *construct* a bit string of length eight that ends with the two bits 00.
- There are $2^5 = 32$ ways to construct such a string.
  - This follows by **the product rule**
    - Because the **first bit** can be chosen in only one way, **each of the second through the sixth bits** can be chosen in two ways, and the **last two bits** can be chosen in one way.
- Consequently, the *number of bit strings of length eight that begin with a 1 or end with a 00 equals* $128 + 64 - 32 = 160$
  - Equals the number of ways to construct a bit string of length eight that begins with a 1 or that ends with 00

Inclusion-Exclusion Principle: Example 2

- **Example 2:**
  - A computer company receives **350** applications from college graduates for a job planning a line of new web servers.
  - Suppose that **220** of these applicants majored in computer science, **147** majored in business, and **51** majored both in computer science and in business.
  - How many of these applicants majored neither in computer science nor in business?
- **Solution:**
  - To find the number of these applicants who majored neither in computer science nor in business, we can **subtract** the number of students who majored either in computer science or in business (or both) from the total number of applicants.

Inclusion-Exclusion Principle: Example 2 (*cont.*)

- **Let**:
  - A_1 be the set of students who majored in computer science.
  - A_2 the set of students who majored in business.
- **Then**:
  - $A_1 \cup A_2$ is the set of students who majored in computer science or business (or both).
  - $A_1 \cap A_2$ is the set of students who majored both in computer science and in business.
- By **the subtraction rule**, the number of students who majored either in computer science or in business (or both) equals:
  - $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2| = 220 + 147 - 51 = 316$
- We **conclude** that:
  - **350 – 316 = 34** of the applicants majored neither in computer science nor in business.
- A Venn diagram for this example is shown in **Figure 3**



$$\begin{aligned}
|\overline{A_1 \cup A_2}| &= |U| - |A_1 \cup A_2| \\
&= |U| - (|A_1| + |A_2| - |A_1 \cap A_2|) \\
&= 350 - (220 + 147 - 51) \\
&= 350 - 316 \\
&= 34
\end{aligned}$$

FIGURE 3 **Applicants who majored in neither computer science nor business.**

The Division Rule

- We have introduced the **product**, **sum**, and **subtraction** rules for **counting**.
- You may wonder whether there is also a **division rule** for **counting**.
- In fact, there is such a rule, which can be useful when solving certain types of enumeration problems.

THE DIVISION RULE There are n/d ways to do a task if it can be done using a procedure that can be carried out in n ways, and for every way w, exactly d of the n ways correspond to way w.

The Division Rule

- We can restate **the division rule** in terms of **sets**:
  - "If the finite set **A** is the union of ***n*** pairwise disjoint subsets each with ***d*** elements, then ***n* = |A| / *d***".
- We can also formulate **the division rule** in terms of **functions**:
  - "If ***f*** is a function from **A** to **B** where **A** and **B** are finite sets, and that for every value ***y*** ∈ **B** there are exactly ***d*** **values x** ∈ **A** such that ***f (x) = y*** (in which case, we say that ***f*** is ***d*-to-one**), then **|B| = |A| / d**"
- **Remark**:
  - **The division rule** comes in handy when it appears that a **task** can be done in ***n*** different ways, but it turns out that for each way of doing the task, there are ***d*** **equivalent** ways of doing it.
  - Under these circumstances, we can conclude that there are **n / d inequivalent** ways of doing the task.
- We illustrate the use of **the division rule** for **counting** with the next **two examples**.

The Division Rule: Examples

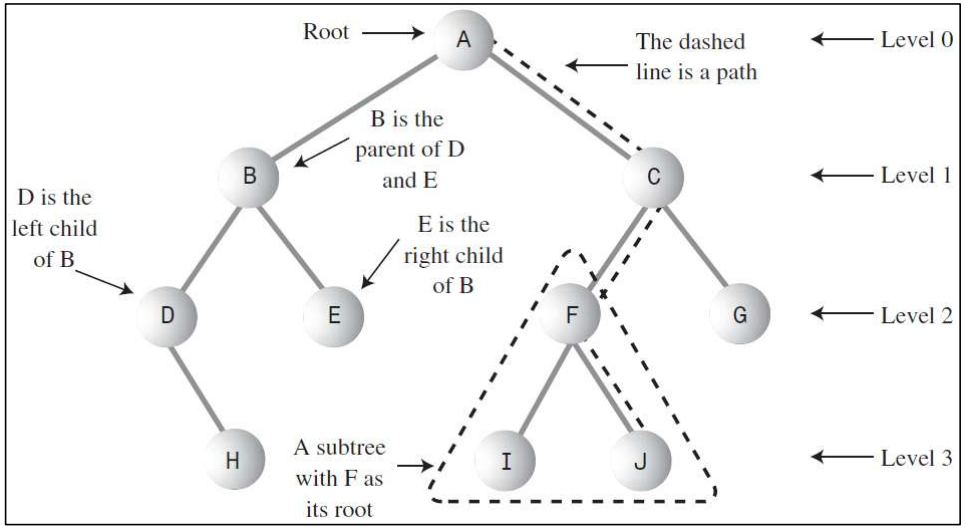
- **Example 1:**
  - Suppose that an automated system has been developed that counts the legs of cows in a pasture.
  - Suppose that this system has determined that in a farmer's pasture there are exactly **572** legs.
  - How many cows are there is this pasture, assuming that each cow has **four** legs and that there are no other animals present?
- **Solution:**
  - Let ***n*** be the number of cow legs counted in a pasture.
  - Because each cow has **four** legs, by **the division rule** we know that the pasture contains **n / 4** cows.
    - Consequently, the pasture with **572** cow legs has **572 / 4 = 143** cows in it.

The Division Rule: Examples

- **Example 2:**
  - How many different ways are there to seat **four** people around a circular table, where **two seatings** are considered the same when each person has the same **left** neighbour and the same **right** neighbour?
- **Solution:**
  - We *arbitrarily* select a seat at the table and label it **seat 1**
  - We number the rest of the seats in numerical order, proceeding *clockwise* around the table.
  - **Note** that are **four** ways to select the person for **seat 1**, **three** ways to select the person for **seat 2**, **two** ways to select the person for **seat 3**, and **one** way to select the person for **seat 4**.
  - Thus, there are **4! = 24** ways to order the given **four** people for these seats.
  - However, each of the **four** choices for **seat 1** leads to the same arrangement, as we distinguish two arrangements only when one of the people has a **different** immediate **left** or immediate **right** neighbour.
  - Because there are **four** ways to choose the person for **seat 1**, by **the division rule** there are **24 / 4 = 6** different **seating arrangements** of **four** people around the **circular table**.

Tree Diagrams

- **Counting** problems can be solved using **tree** diagrams.
- A **tree** consists of:
  - A **root**
  - A number of **branches** leaving the **root**
  - possible **additional branches** leaving the endpoints of other branches.
- To use trees in counting:
  - We use a branch to represent each possible choice.
  - We represent the **possible outcomes** by the **leaves**, which are the **endpoints of branches not having other branches starting at them**.



Tree Diagrams: Example

- **Example:**
  - How many bit strings of length four do not have two consecutive 1s?
- **Solution:**
  - The tree diagram in **Figure 4** displays all bit strings of length four without two consecutive 1s.
  - We see that there are eight bit strings of length four without two consecutive 1s.

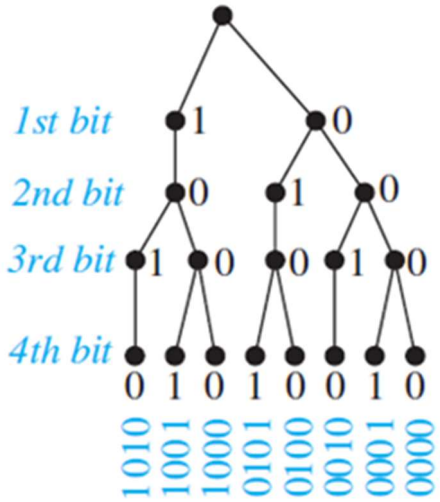


FIGURE 4 Bit strings of length four without consecutive 1s.

The Pigeonhole Principle: Introduction

- Suppose that a flock of **20 pigeons** flies into a set of **19 pigeonholes** to roost.
- Because there are **20 pigeons** but only **19 pigeonholes**:
  - **A least one** of these **19 pigeonholes** must have **at least two pigeons** in it.
- To see why this is true, note that if each pigeonhole had at most one pigeon in it, at most 19 pigeons, one per hole, could be accommodated.
- This illustrates a general principle called the **pigeonhole principle**, which states that if there are more pigeons than pigeonholes, then there must be at least one pigeonhole with at least two pigeons in it (see **Figure 1**).
- This **principle** is extremely useful;
  - it applies to much more than pigeons and pigeonholes.



(a) (b) (c)

FIGURE 1 **There are more pigeons than pigeonholes.**

- **Theorem 1**:

THE PIGEONHOLE PRINCIPLE If k is a positive integer and $k + 1$ or more objects are placed into k boxes, then there is at least one box containing two or more of the objects.

The Pigeonhole Principle: Examples

- **Corollary 1**:

> A function f from a set with $k + 1$ or more elements to a set with k elements is not one-to-one.

- **Example 1:**
  - Among any group of **367 people**, there must be at least two with the **same birthday**
    - Because there are only **366** possible **birthdays**.
- **Example 2:**
  - In any group of **27 English words**, there must be at least two that begin with the same letter
    - Because there are **26 letters** in the **English alphabet**.

The Pigeonhole Principle: Examples

- **Example 3:**
  - How many students must be in a class to guarantee that at least two students receive the same score on the final exam, if the exam is graded on a scale from **0** to **100** points?
- **Solution:**
  - There are **101** possible scores on the final.
  - The **pigeonhole principle** shows that among any **102** students there must be at least 2 students with the same score.

The Generalized Pigeonhole Principle

- The **pigeonhole principle** states that there must be at least two objects in the same box when there are more objects than boxes.
- **However**, even more can be said when the number of objects exceeds a multiple of the number of boxes.
- **For instance**, among any set of **21 decimal digits** there **must be 3 that are the same**.
- This follows because when **21 objects** are distributed into **10 boxes**, **one box must have more than 2 objects**.
- **Theorem 2**:

THE GENERALIZED PIGEONHOLE PRINCIPLE If N objects are placed into k boxes, then there is at least one box containing at least $\lceil N/k \rceil$ objects.

The Generalized Pigeonhole Principle: Examples

- **Example 1:**
  - Among **100** **people** there are at least $\lceil 100/12 \rceil = 9$ who were born in the same month.
- **Example 2:**
  - What is the **minimum** number of students required in a discrete mathematics class to be sure that **at least** **six** will receive the **same grade**, if there are **five** possible grades, **A**, **B**, **C**, **D**, and **F**?
- **Solution:**
  - The **minimum** number of students needed to ensure that at **least** **six** **students** receive the **same grade** is the **smallest integer** N such that $\lceil N/5 \rceil = 6$
  - The **smallest** such integer is $N = 5 \cdot 5 + 1 = 26$
  - If you have **only** **25** **students**, it is possible for there to be **five** who have received each grade so that **no** **six** **students** have received the **same grade**.
  - Thus, **26** is the **minimum** number of students needed to ensure that at least **six** students will receive the **same grade**.

Any Questions?

- Have a look at the "**Extra Examples**" on the textbook's website.
- Keep reviewing this lecture *regularly, as well as other lectures*.
- Have a look at the **exercises** at the end of the sections we covered in this lecture in **Chapter 6** and do your best with those ones similar to what we presented in this lecture.
  - You should get more *examples* and *exercises* in the *tutorials*.
- Ask *me or ask your peers* if you have any questions.
  - Please, do not wait until the end of the semester or even right before exams.

All the best!