Ceng 124
Discrete Structures

2018-2019 Spring Semester

by Sibel T. Özyer, Spring 2019

Topics

- 2.3 Functions
- 2.4 Sequences and Summations

Topics

- 2.3 Functions
  - Definition of a Function.
    - Domain, Codomain
    - Image, Preimage
  - Injection, Surjection, Bijection
  - Inverse Function
  - Function Composition
  - Graphing Functions
  - Floor, Ceiling Functions
- 2.4 Sequences and Summations
  - Sequences.
    - Geometric Progression, Arithmetic Progression
  - Recurrence Relations
  - Summations

Functions

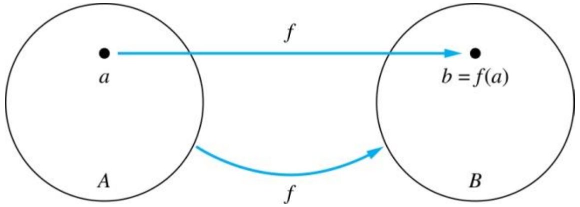
- Definition: Let A and B be nonempty sets. A function f from A to B, denoted $f: A \to B$ is an assignment of each element of A to exactly one element of B. We write $f(a) = b$ if b is the unique element of B assigned by the function f to the element a of A.
- Functions are sometimes called *mappings* or *transformations*.

Domain-Codomain-Image-Preimage

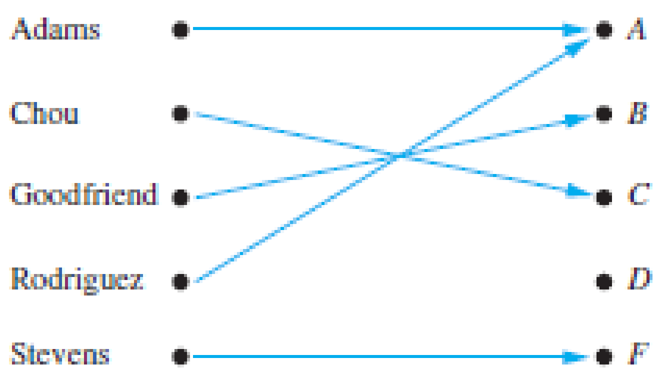
Given a function $f: A \to B$:

- We say f *maps* A to B *or*

  f is a *mapping* from A to B.
- A is called the *domain* of f.
- B is called the *codomain* of f.
- If $f(a) = b$,
  - then b is called the *image* of a under f.
  - a is called the *preimage* of b.
- The range of f is the set of all images of points in **A** under f. We denote it by $f(\mathbf{A})$.
- Two functions are *equal* when they have the same domain, the same codomain and map each element of the domain to the same element of the codomain.



Example



Assignment of Grades

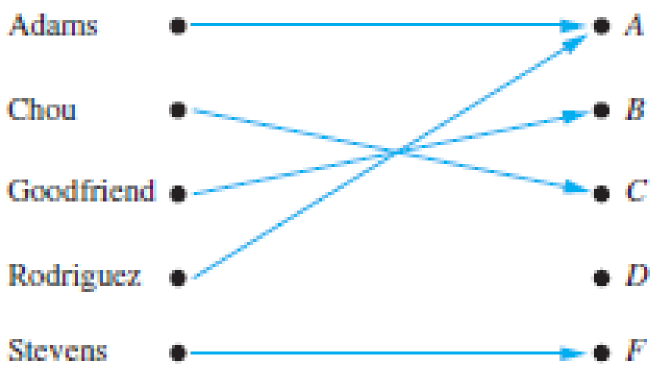
- Let G is the function name
- Domain of the set G

  {Adams, Chou, Goodfriend, Rodriguez, Stevens}
- Codomain is the set {A,B,C,D, F}.
- The range of G is the set {A,B,C, F},

  because each grade except D is assigned to some student.

Representing Functions

- Functions are specified in many different ways.
  - An explicit statement of the assignments, as in Figure.
  - A formula, such as $f(x) = x + 1$, to define a function.
  - A computer program to specify a function.



Assignment of Grades

Functions in Programming Languages

- The domain and codomain of functions are often specified in programming languages. For instance, the Java statement

```
int floor(float real)  {. . .}
```

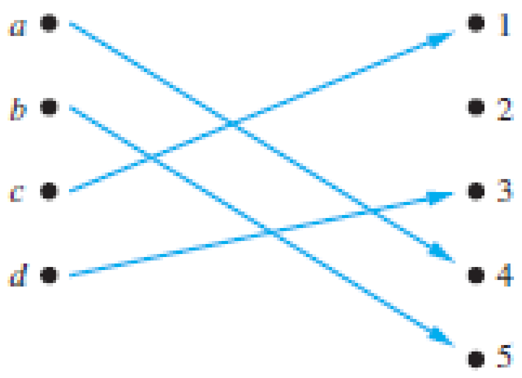
- and the C++ function statement

```
int function (float x)  {. . .}
```

- both tell us that the *domain* of the floor function is the set of *real numbers* (represented by floating point numbers) and its *codomain* is the set of *integers*.

One-to-One Function

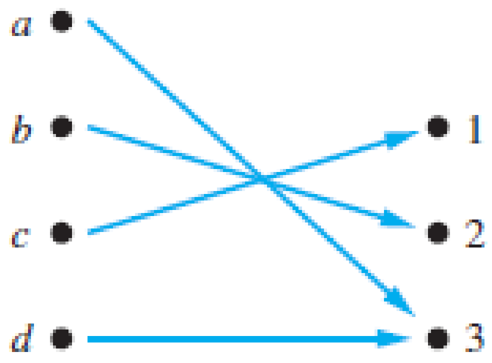
- Definition: A function f is said to be *one-to-one* , or *injective*, if and only if $f(a) = f(b)$ implies that $a = b$ for all a and b in the domain of f. A function is said to be an *injection* if it is one-to-one.



An one-to-one function

Onto Function

- **Definition:** A function f from A to B is called *onto* or *surjective*, if and only if for every element $b \in B$ there is an element $a \in A$ with $f(a) = b$ A function f is called a surjection if it is onto.

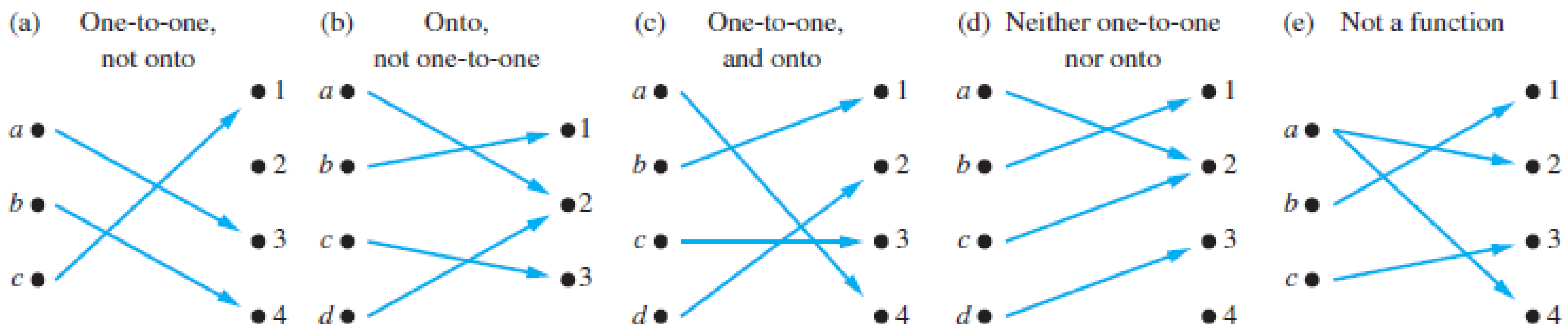


An onto Function

One-to-one Correspondence

- Definition: A function f is a *one-to-one correspondence*, or a ***bijection***, if it is both one-to-one and onto (surjective and injective).

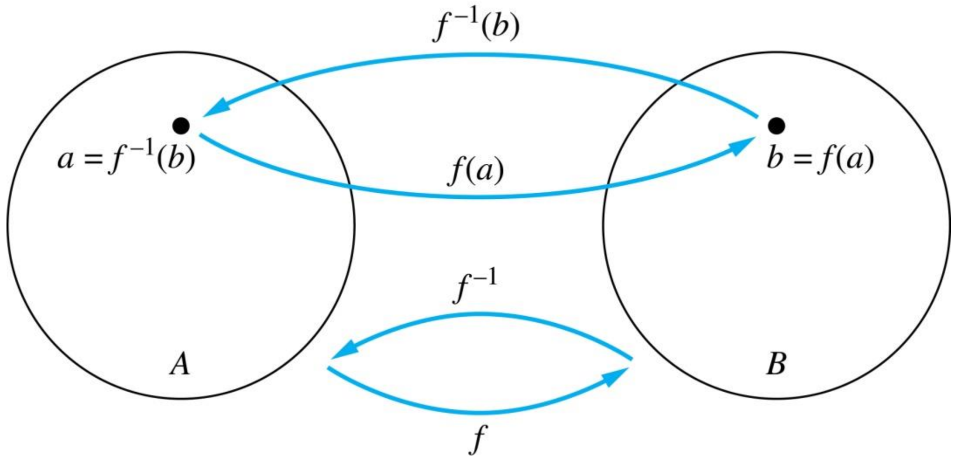
Different Types of Correspondences



Inverse Function

- Definition: Let f be a bijection from A to B. Then the inverse of f, denoted f^{-1} , is the function from B to A defined as

$$f^{-1}(y) = x \text{ iff } f(x) = y$$



Inverse Question

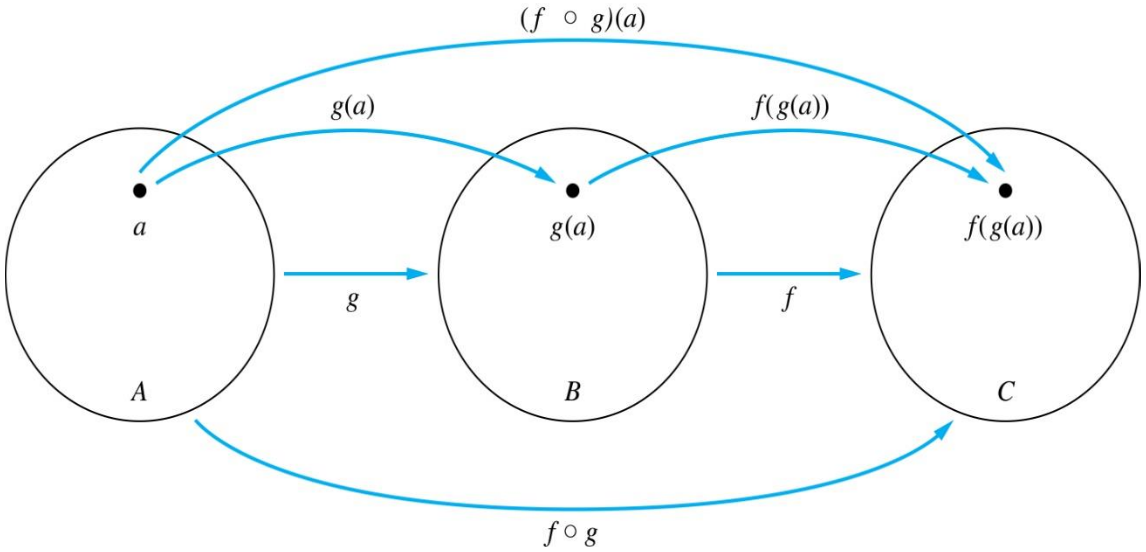
Example: Let f be the function from $\{a, b, c\}$ to $\{1, 2, 3\}$ such that $f(a) = 2$, $f(b) = 3$, and $f(c) = 1$. Is f invertible and if so what is its inverse?

Solution: The function f is invertible because it is a one-to-one correspondence. The inverse function f^{-1} reverses the correspondence given by f, so $f^{-1}(1) = c$, $f^{-1}(2) = a$, and $f^{-1}(3) = b$.

Composition

- **Definition**: Let $f: B \to C$, $g: A \to B$. The *composition of f with g*, denoted $f \circ g$ is the function from A to C defined by

$$f \circ g(x) = f(g(x))$$



Composition Question

- Example: Let f and g be functions from the set of integers to the set of integers defined by $f(x) = 2x + 3$ and $g(x) = 3x + 2$.

  What is the composition of f and g, and also the composition of g and f ?

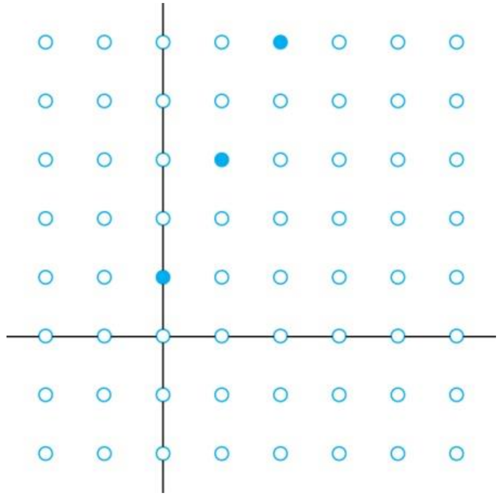
- Solution:

  $f \circ g\ (x) = f(g(x)) = f(3x + 2) = 2(3x + 2) + 3 = 6x + 7$

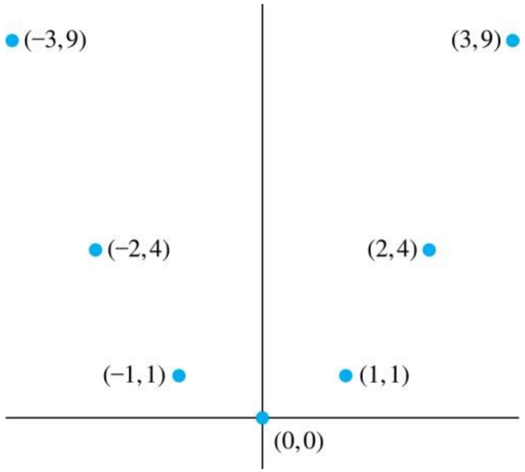
  $g \circ f\ (x) = g(f(x)) = g(2x + 3) = 3(2x + 3) + 2 = 6x + 11$

Graphs of Functions

- Let f be a function from the set A to the set B. The graph of the function f is the set of ordered pairs {(a,b) | a ∈A and f(a) = b}.



Graph of f(n)=2n+1 from Z to Z



Graph of f(x)= x^2 from Z to Z

Floor and Ceiling Functions

- Floor and Ceiling functions are two important functions in discrete mathematics.
- The *floor* function, denoted $$f(x) = \lfloor x \rfloor$$

  is the largest integer less than or equal to *x*.

- The *ceiling* function, denoted $$f(x) = \lceil x \rceil$$

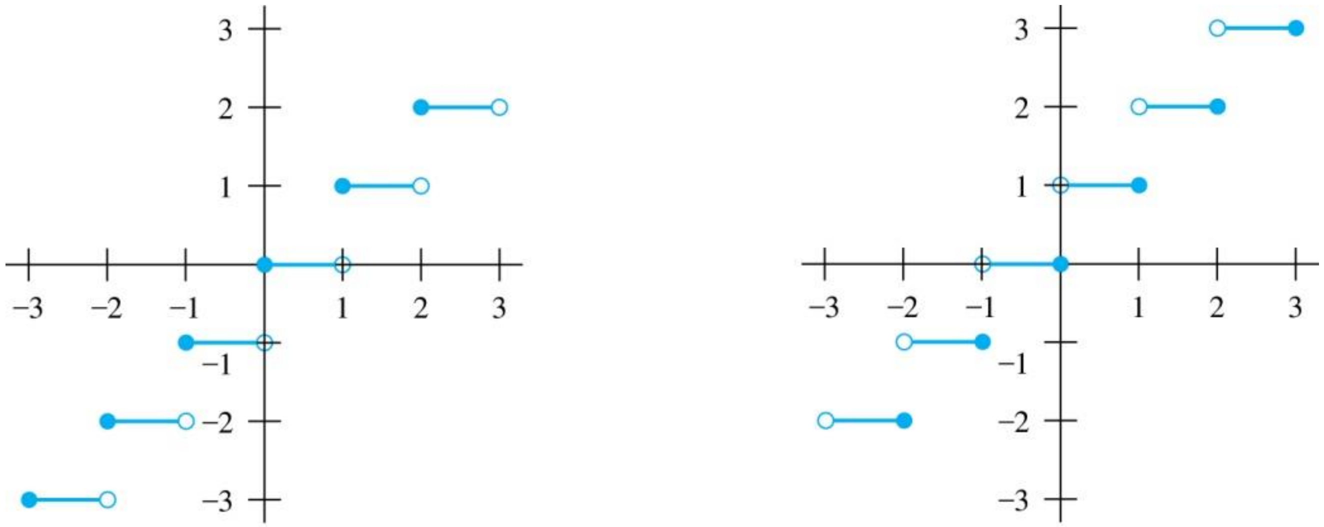
  is the smallest integer greater than or equal to *x*

- Examples:

$$\lceil 3.5 \rceil = 4 \qquad \lfloor 3.5 \rfloor = 3$$

$$\lceil -1.5 \rceil = -1 \qquad \lfloor -1.5 \rfloor = -2$$

Graphs of Floor and Ceiling Functions



(a) $y = [x]$ (b) $y = [x]$

Graph of (a) Floor and (b) Ceiling Functions

Question1

- Data stored on a computer disk or transmitted over a network are usually represented as a string of bytes. Each byte is made up of 8 bits. How many bytes are required to encode 100 bits of data?

Solution: $\lceil 100/8 \rceil = \lceil 12.5 \rceil = 13$ bytes are required.

Question2

- In synchronous transfer mode(ATM) (a communication protocol used on backbone networks), data are organized into cells of 53 bytes. How many ATM cells can be transmitted in 1 minute over a connection that transmits data at the rate of 500 kilobits per second?

Solution: In 1 minute, this connection can transmit

500.000 * 60=30.000.000 bits

Each ATM cell is 53 bytes long, means that it is 53*8=424 bits long.

$\lfloor 30,000,000/424 \rfloor = 70,754$ ATM cells can be transmitted in 1 minute over a

500 kilobit per sec connection.

2.4 Sequences and Summations

- Sequences are ordered lists of elements.
  - 1, 2, 3, 5, 8
  - 1, 3, 9, 27, 81, …….
- Used in discrete mathematics.
- An important data structure in computer science.
- We will introduce the terminology to represent sequences and sums of the terms in the sequences.

Sequences

Definition: A *sequence* is a function from a subset of the integers (usually either the set $\{0, 1, 2, 3, 4, \ldots..\}$ or $\{1, 2, 3, 4, \ldots.\}$) to a set S.

- The notation a_n is used to denote the image of the integer n. We can think of a_n as the equivalent of $f(n)$ where f is a function from $\{0,1,2,\ldots..\}$ to S. We call a_n a *term* of the sequence.

Sequences (cont.)

- **Example:** Consider the sequence $\{a_n\}$ where

$$a_n = \frac{1}{n} \qquad \{a_n\} = \{a_1, a_2, a_3, \ldots\}$$

$$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \ldots$$

Geometric Progression

Definition: A *geometric progression* is a sequence of the form:

$$a, ar, ar^2, \ldots, ar^n, \ldots$$

where the *initial term a* and the *common ratio r* are real numbers.

Examples:

1. Let $a = 1$ and $r = -1$. Then:

$$\{b_n\} = \{b_0, b_1, b_2, b_3, b_4, \ldots\} = \{1, -1, 1, -1, 1, \ldots\}$$

1. Let $a = 2$ and $r = 5$. Then:

$$\{c_n\} = \{c_0, c_1, c_2, c_3, c_4, \ldots\} = \{2, 10, 50, 250, 1250, \ldots\}$$

2. Let $a = 6$ and $r = 1/3$. Then:

$$\{d_n\} = \{d_0, d_1, d_2, d_3, d_4, \ldots\} = \{6, 2, \frac{2}{3}, \frac{2}{9}, \frac{2}{27}, \ldots\}$$

Arithmetic Progression

Definition: A *arithmetic progression* is a sequence of the form:

$$a, a+d, a+2d, \ldots, a+nd, \ldots$$

where the *initial term a* and the *common difference d* are real numbers.

Examples:

1. Let $a = -1$ and $d = 4$:

$$\{s_n\} = \{s_0, s_1, s_2, s_3, s_4, \ldots\} = \{-1, 3, 7, 11, 15, \ldots\}$$

2. Let $a = 7$ and $d = -3$:

$$\{t_n\} = \{t_0, t_1, t_2, t_3, t_4, \ldots\} = \{7, 4, 1, -2, -5, \ldots\}$$

3. Let $a = 1$ and $d = 2$:

$$\{u_n\} = \{u_0, u_1, u_2, u_3, u_4, \ldots\} = \{1, 3, 5, 7, 9, \ldots\}$$

Strings

Definition: A *string* is a finite sequence of characters from a finite set (an alphabet).

- Sequences of characters or bits are important in computer science.
- The *empty string* is represented by λ.
- The string *abcde* has *length* 5.

Recurrence Relations

Definition: A *recurrence relation* for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for all integers n with $n \geq n_0$, where n_0 is a nonnegative integer.

- A sequence is called a *solution* of a recurrence relation if its terms satisfy the recurrence relation.
- The *initial conditions* for a sequence specify the terms that precede the first term where the recurrence relation takes effect.

Questions about Recurrence Relations

Example 1: Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 1,2,3,4,\ldots$. and suppose that $a_0 = 2$. What are a_1, a_2 and a_3?
[Here $a_0 = 2$ is the initial condition]

Solution: We see from the recurrence relation that

$$a_1 = a_0 + 3 = 2 + 3 = 5$$
$$a_2 = 5 + 3 = 8$$
$$a_3 = 8 + 3 = 11$$

Questions about Recurrence Relations

Example 2 : Let $\{a_n\}$ be a sequence that satisfies the recurrence relation

$a_n = a_{n-1} - a_{n-2}$ for $n = 2,3,4,\ldots.$ and suppose that $a_0 = 3$ and $a_1 = 5$.

What are a_2 and a_3?

[Here the initial conditions are $a_0 = 3$ and $a_1 = 5$.]

Solution: We see from the recurrence relation that

$$a_2 = a_1 - a_0 = 5 - 3 = 2$$

$$a_3 = a_2 - a_1 = 2 - 5 = -3$$

Sequences Table

TABLE 1 Some Useful Sequences.	
nth Term	*First 10 Terms*
n^2	1, 4, 9, 16, 25, 36, 49, 64, 81, 100, ...
n^3	1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, ...
n^4	1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000, ...
2^n	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ...
3^n	3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049, ...
$n!$	1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, ...
f_n	1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Summations

- Sum of the terms $a_m, a_{m+1}, \ldots, a_n$
- The notation:

$$\sum_{j=m}^{n} a_j \quad \textstyle\sum_{j=m}^{n} a_j \quad \sum_{m \le j \le n} a_j$$

represents

$$a_m + a_{m+1} + \cdots + a_n$$

- The variable *j* is called the *index of summation*. It runs through all the integers starting with its *lower limit* *m* and ending with its *upper limit* *n*.

Summations(cont.)

- More generally for a set *S*:

$$\sum_{j \in S} a_j$$

- Examples:

$$r^0 + r^1 + r^2 + r^3 + \cdots + r^n = \sum_0^n r^j$$

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots = \sum_1^\infty \frac{1}{i}$$

$$\text{If } S = \{2, 5, 7, 10\} \text{ then } \sum_{j \in S} a_j = a_2 + a_5 + a_7 + a_{10}$$

Index of Summation

- Suppose we have the sum $\sum_{j=1}^{5} j^2$, Indexes are 1,2,3,4,5.

$$\sum_{j=1}^{5} j^2 = \sum_{k=0}^{4} (k+1)^2$$

- Both sums are 1+4+9+16+25 = 55

Double Summation

- Nested loops in computer programs. An example: $\sum_{i=1}^{4}\sum_{j=1}^{3} ij.$
- Evaluation of the double sum:

$$\sum_{i=1}^{4}\sum_{j=1}^{3} ij = \sum_{i=1}^{4}(i + 2i + 3i)$$

$$= \sum_{i=1}^{4} 6i$$

- 6+12+18+24 = 60

Summation Formulas

TABLE 2 Some Useful Summation Formulae.

Sum	*Closed Form*
$\sum_{k=0}^{n} ar^k \ (r \neq 0)$	$\frac{ar^{n+1} - a}{r - 1}, r \neq 1$
$\sum_{k=1}^{n} k$	$\frac{n(n+1)}{2}$
$\sum_{k=1}^{n} k^2$	$\frac{n(n+1)(2n+1)}{6}$
$\sum_{k=1}^{n} k^3$	$\frac{n^2(n+1)^2}{4}$
$\sum_{k=0}^{\infty} x^k, \lvert x \rvert < 1$	$\frac{1}{1-x}$
$\sum_{k=1}^{\infty} kx^{k-1}, \lvert x \rvert < 1$	$\frac{1}{(1-x)^2}$