Data Structures And Algorithms

Tutorial One:

Introduction to Primitive Data Types

<u>Lecture Instructor</u> : Prof. Dr. Ahmed El Nahas

<u>Tutorial Instructors</u>:

Eng. Ahmed Abdelhamid Waheed Abdelwahab

Eng. Magdy Sedra

Quick Review: Algorithm

- A technique, procedure, process, routine or set of rules giving a sequence of operations for solving a specific type of problem.
  - Some problems have no algorithm
    - Example: find integers a, b, c, n > 2 such that "$a^n + b^n = c^n$".
  - Other problems may have one or more algorithms
    - Example: sorting elements in an array.
      - **Selection Sort**
      - **Bubble Sort**
    - Example: searching for elements in an array.
      - **Linear Search**
      - **Binary Search**

Quick Review: Algorithm Procedure

❖A sequence of steps can be considered an algorithm if it satisfies the following features:

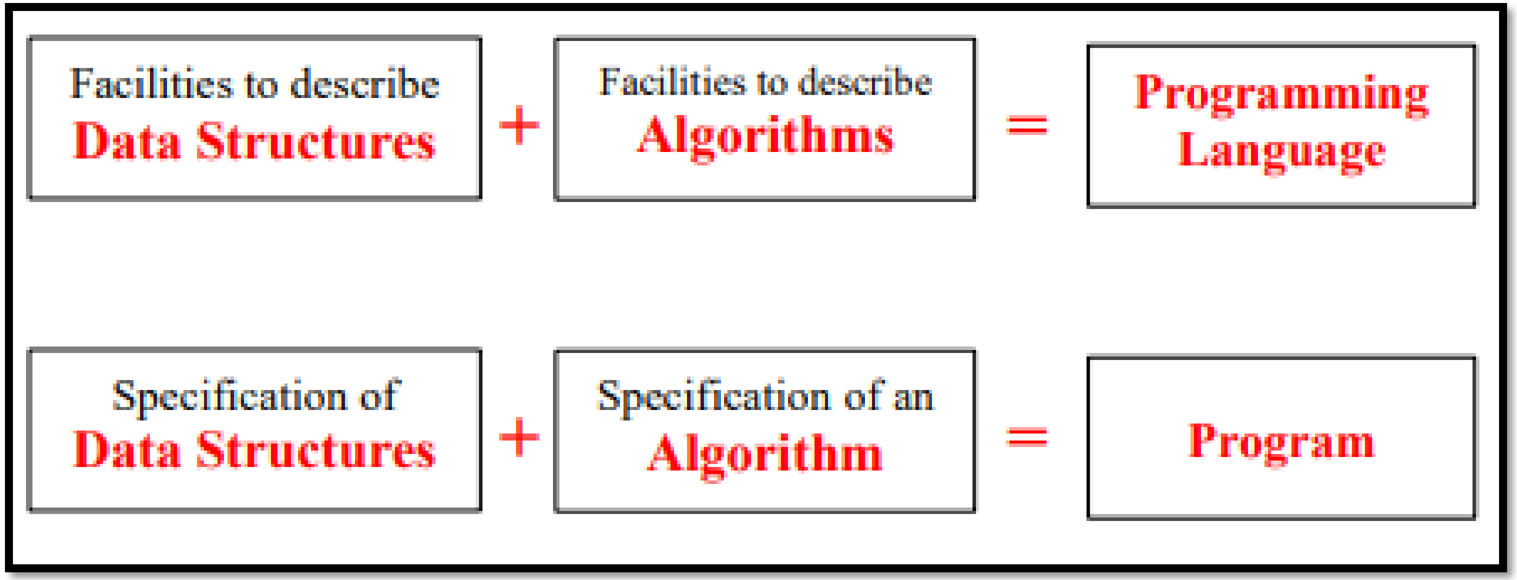
- φ Be finite (is terminating)
- φ Be definite (be unambiguous)
- φ Be effective (the basic operations are primitive that they can be carried out in reasonable time)
- φ Have one or more inputs (to specify an instance of the problem)
- φ Have one or more output.

Quick Review: Data

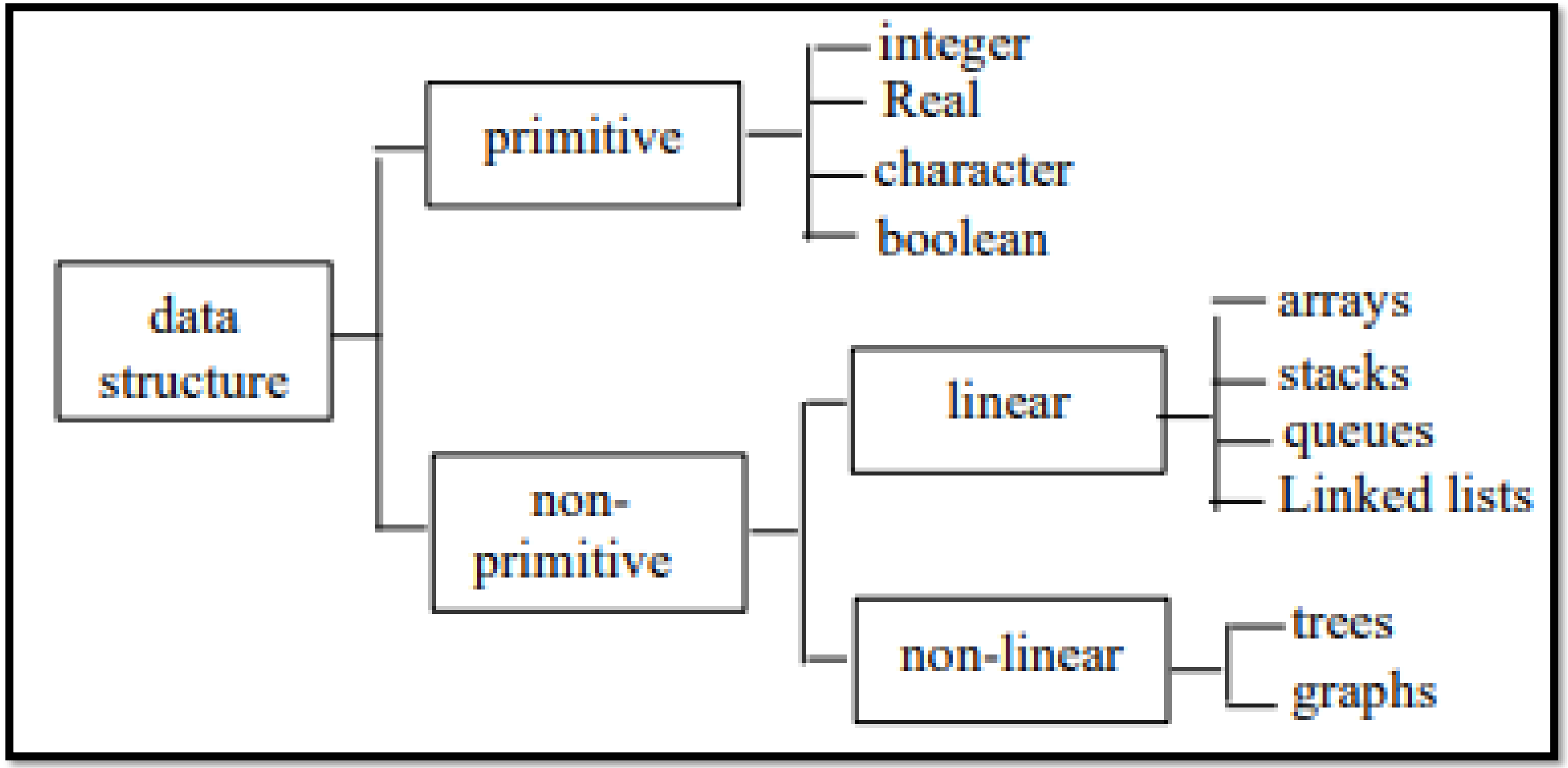
- Data types can be considered as simple or structured.
  - Simple
    - Data item that can store only one value at a time.
    - Some times called **Primitive Data Structure**.
  - Structured
    - Each data item is a collection of other data items.
    - Some times called **Non-Primitive Data Structure**.

Quick Review: Data Structure

❖Data Structure is a collection of related data and a set of rules for organizing and accessing it.



Quick Review: Classification of Data Structures



Quick Review: Integer Data

- The set of integer values defined for this type forms an ordered subset within finite bounds of the infinite set of integers studied in mathematics.
  - Arithmetic operations on integer values will be performed CORRECTLY by the computer if all operands, and all the intermediate results lie in the range of integer data.
  - If the allowable range is exceeded, the result of the arithmetic operation is unpredictable.
    - This called **overflow**.

Quick Review: Representation of Integer Data

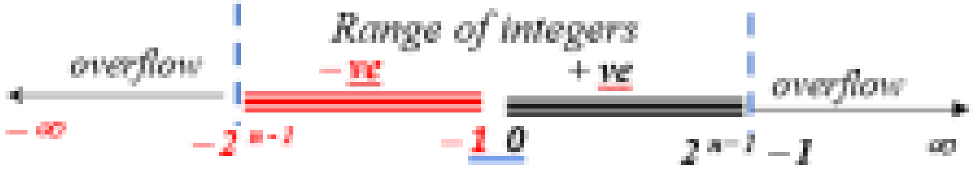
1. Binary Integer (Sign and Magnitude)
2. Binary Integer (2’s Complement)
3. Binary Coded Decimal (BCD) packed.

Quick Review: Binary Integer (Sign and Magnitude)

- For an n-bit representation:
  - Assign the leftmost (most significant) bit to be the sign bit.
    - If the sign bit is 0, this means the number is positive.
    - If the sign bit is 1, then the number is negative.
  - The remaining n-1 bits are used to represent the magnitude of the binary number in the unsigned binary notation.
  - **Range**: $-(2^{n-1} - 1) \leq X \leq 2^{n-1} - 1$

Quick Review: Binary Integer (2’ Complement)

Range of integers that can be represented using n-bits is:



$$-2^{n-1} \leq \text{integer} \leq +(2^{n-1}-1)$$

A negative value X, where: $-2^{n-1} \leq X \leq -1$

is represented by : $2^n - X$

- Positive numbers are computed as they are.
- Negative numbers are computed by:
    1. Get ones complement (negation) of positive corresponding number.
    2. Add one to the result in (1).

Quick Review: Binary Coded Decimal (BCD Packed)

d_n	. . . .	d_1	d_0	*sign*

- Four bits are used for each decimal digit.
- The sign is written on the rightmost end using four bits (usually, 1100 for positive value, 1101 for negative)
- This representation is less efficient with respect to storage requirements. An increase of 4-bits in the size allocated for integer data results in 10-fold range when using *BCD* (one extra decimal digit), but results in 16-fold range when using *binary* representation ($2^4 = 16$)

Quick Review: Logical (Boolean) Data

- ❖Assumes two values (true / false)
- ❖C has no separate data type for logical data. Type integer is overloaded to represent logical data too.
  - φ A zero value represents false
  - φ A non-zero value represents true.
- ❖Short-circuit and, or Operations
  - φ and (&&) or (||) are short circuit operations.
    - ➢ With this form of operation, the right operand is evaluated only if it has to be.
  - φ Simplify coding operations
    - ➢ **If** he has a job <u>*and*</u> his salary > 2000 **then** ....
    - ➢ **If** not end of list <u>*and*</u> next item = X **then** ....
    - ➢ **If** x = 20 <u>*or*</u> x = 10 **then** ....

Quick Review: Character Data

Character data denotes a finite, ordered set of characters. The most widely used character sets are:

- *ASCII* character set (American Standard Code for Information Interchange)
- *EBCDIC* Character set (Extended Binary-Coded-Decimal Interchange Code)
- *Unicode*

A, B, C, D, *, #

A character set is specified by:

- Characters included
- Order of characters in the set
- Code for each character

The characters in a set are classified into:

- Printable characters
- Non-printable (control) characters

Type *char* in C++ is one of *integer* type. Any variable of type *char* may be used in *integer* expression.

- **Backspace**
- **Linefeed**

Exercises(1)

1. How would you interpret the 8-bit signed integer 0101 1101 in each of the following systems
   a) Sign and magnitude
   b) Two's complement
   c) Binary-coded decimal

Solution

1. How would you interpret the 8-bit signed integer 0101 1101 in each of the following systems
   a) Sign and magnitude

      Solution: (Positive Number) $1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 = 93$
   b) Two's complement

      Solution: (Positive Number) $1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 = 93$
   c) Binary-coded decimal

      Solution: 1101 → negative

      0101 → 5

      -5

Exercises(2)

2. Using two’s complement what is the range of integer data that can be represented in one byte (8-bit)?

Solution

2. Using two's complement what is the range of integer data that can be represented in one byte (8-bit)?

Solution: Largest: $2^7 - 1 = 127$

Smallest: $-2^7 = -128$

$-128 \leq X \leq 127$

Exercises(3)

3. Using 24-bit, what are the largest and smallest integers that can be represented in:
   a) Signed magnitude binary integer
   b) Two’s complement notation
   c) BCD ?

Solution(a)

$2^{23} - 1 = 8388607$ [Largest]

$-(2^{23} - 1) = -8388607$ [Smallest]

Solution(b)

$2^{23} - 1 = 8388607$ [Largest]

$-(2^{23}) = -8388608$ [Smallest]

Solution(c)

4 bits sign

20 bits value

20/4 digits → 5 decimal digits

99999 [Largest]

-99999 [Smallest]

Exercises(4)

4. Integer data in a certain computer are represented using two’s complement in 16-bit word;
   a) What is the range of integer data that can be represented ?
   b) Give the representation of the following integers:
      i. -255
      ii. -27
      iii. 13

Solution(a)

- Range:

  $-2^{15} \leq X \leq 2^{15}-1 \rightarrow -32769 \leq X \leq 32767$

Solution(b)

i. -255

255

255															
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Ones complement of 255															
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Two’s complement of 255															
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1

Solution(b)

ii. -27

27

27															
0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1

Ones complement of 255															
1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0

Two’s complement of 255															
1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1

Solution(b)

iii. 13

13															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Exercises(5)

5. In what ways is it less efficient to represent integers as packed (BCD) decimal rather than as binary integers? Approximately how many bits would be required to represent values as large as 1,000,000,000,000 in each of the two representation?

Solution

This representation is less efficient with respect to storage requirements. An increase of 4-bits in the size allocated for integer data results in 10-fold range when using *BCD* (one extra decimal digit), but results in 16-fold range when using *binary* representation ($2^4 = 16$)

Binary → $\lceil \log_2 1,000,000,000,000 \rceil = 40\ bit + 1\ bit\ sign$ = 41 bits
BCD → 13 digits x 4 bits/digit = 52 bits + 4 bits sign = 56 bits

Exercises(6)

6. Express the following values in packed decimal storage representation (BCD) (**Hint**: use 16-bit format).
   a) 27
   b) 7
   c) -7
   d) 256

Solution(a)

0	2	7	+ve
0000	0010	0111	1100

Solution(b)

0	0	7	+ve
0000	0000	0111	1100

Solution(c)

0	0	7	-ve
0000	0000	0111	1101

Solution(d)

2	5	6	+ve
0010	0101	0111	1101