

SCJ2013 Data Structure & Algorithms

Introduction to Data Structures & Algorithm

Nor Bahiah Hj Ahmad & Dayang
Norhayati A. Jawawi

Objectives:

By the end of the class, students are expected to understand the following:

- problem solving introduction
- algorithm **concept**
- data structure **concept**

Software Eng. & Problem Solving

- Software engineering
 - Provides **techniques to facilitate** the development of computer program
- Problem solving
 - Taking the **statement of a problem** and **develop a computer program** to solve problems.
 - The entire process requires to pass **many phases**, from understanding the problem, design solution and implement the solution.

Problem Solving

- A **solution** to a problem is computer program written in programming language which consist of **modules**.
- Type of **Modules**:
 - A single, stand-alone **function**
 - A **method** of a class
 - A **class**
 - **Several functions or classes** working closely together
 - Other blocks of code

Problem Solving

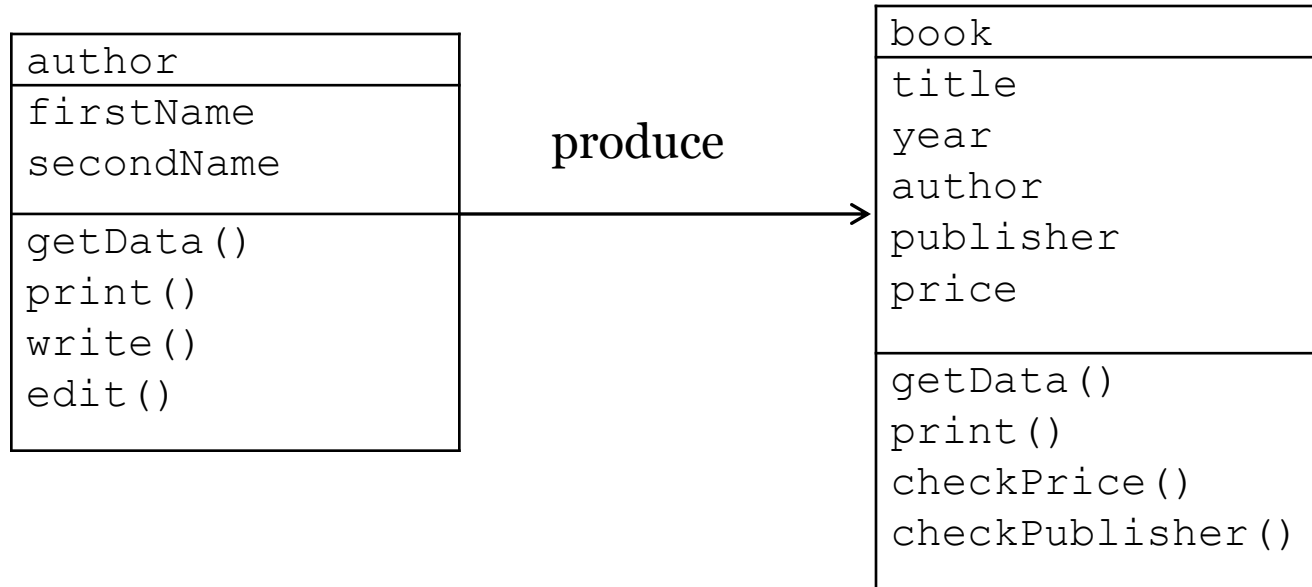
A **good solution** consists of :

- **Modules** that
 - organize **data collection** to facilitate operations
 - must store, move, and alter data
 - use algorithms to communicate with one another

Modularity

- Advantage of module:
 - Constructing programs – small/large modules
 - Debugging programs – task of debugging large program is reduced to small modular program.
 - Reading programs- easier to understand compared to large program
 - Modifying programs – reduce large modification by concentrating on modules
 - Eliminating redundant code – by calling the modules will avoid the same code to be written multiple times

Modularity Example



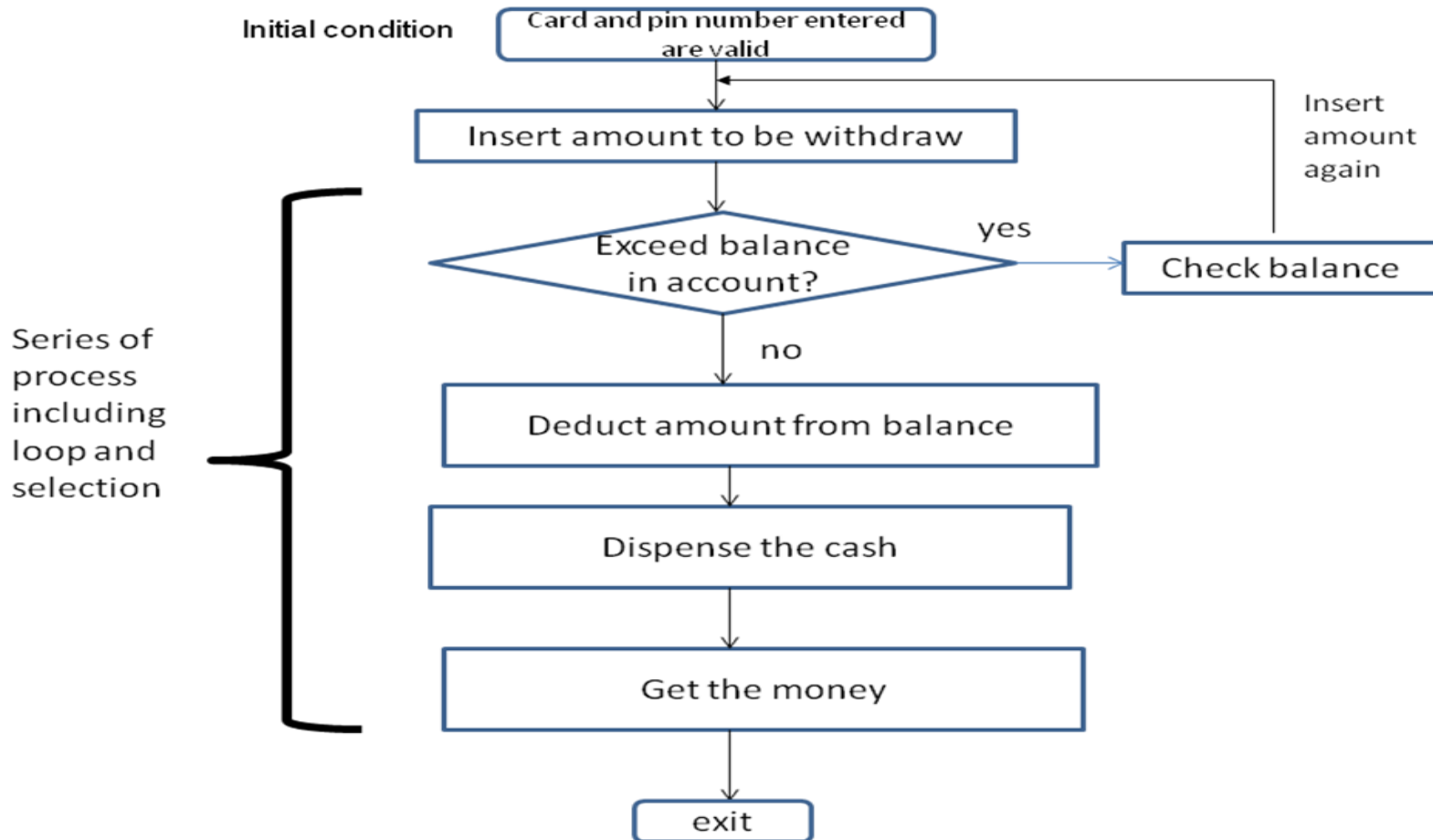
Algorithm

- Module implements algorithms
 - **Algorithm**: a **step-by-step** recipe for performing a task within a finite period of time
 - **Algorithms** often **operate on a collection of data**, which is stored in a structured way in the computer memory (**Data Structure**)
 - **Algorithms**: Problem solving using **logic**

Algorithm

- Well-defined instructions in algorithm includes:
 1. when given an initial state, (INPUT)
 2. proceed through a well-defined series of successive states, (PROCESS)
 3. eventually terminating in an end-state (OUTPUT)

Algorithm



Algorithm

- **3 types of algorithm basic control structure**
 - Sequential
 - Selection
 - Repeation (Looping)

Algorithm

- **Basic algorithm characteristics**
 - Finite solution
 - Clear instructions
 - Has input to start the execution
 - Has output as the result of the execution
 - Operate effectively
- **Algorithm creation techniques**
 - Flowchart, pseudo code, language etc
- **Factors for measuring good algorithm**
 - Running time
 - Total memory usage

Algorithm & Data Structure

– Data Structure

- A way of storing and organizing data in a computer so that it can be used efficiently
- Choosing the right data structure will allow the most efficient algorithm to be used
- A well-designed data structure :
 - allows a variety of critical operations to be performed
 - unable to use few resources, both execution time and memory space, as possible

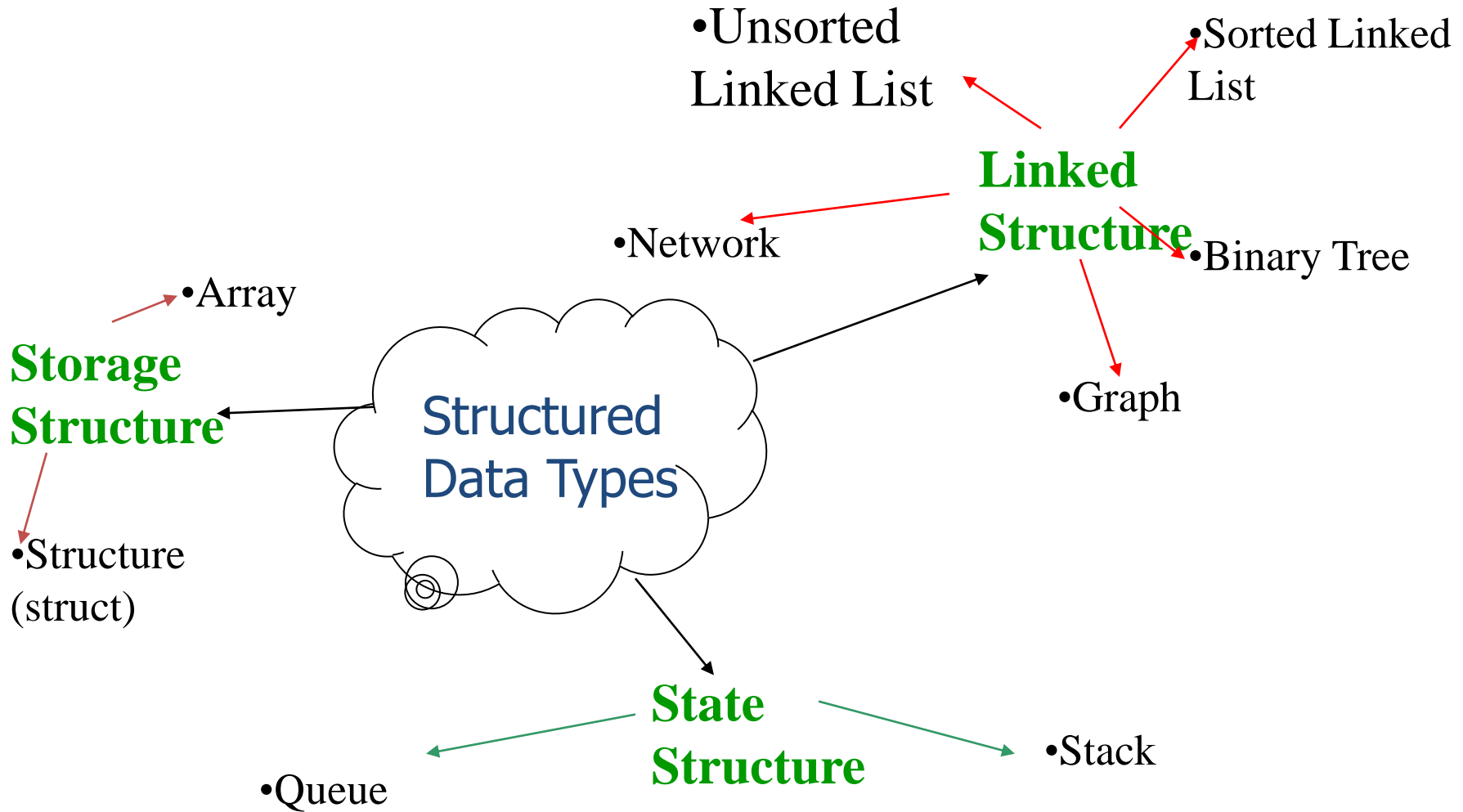
Data Structure

- **Operations to the Data Structure**
 - Traversing- access and process every data in data structure at least once
 - Searching – search for a location of data
 - Insertion – insert item in the list of data
 - Deletion - delete item from a set of data
 - Sorting – sort data in certain order
 - Merging – merge multiple group of data

Data Types

- 2 data types
 1. Basic data types and
 2. structured data types
- Basic Data Types (C++) – store only a single data
 - Integral
 - Boolean – bool
 - Enumeration – enum
 - Character - char
 - Integer – short, int, long
 - Floating point – float, double

Data Types



Data Types

- Structured Data Types
 - Array – can contain multiple data with the same types
 - Struct – can contain multiple data with different type

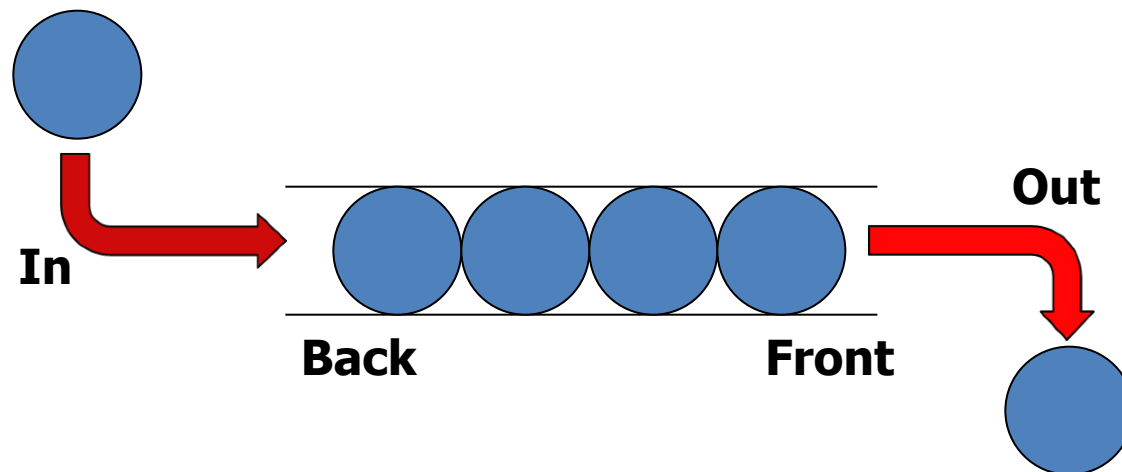
```
typedef struct {  
    int age;  
    char *name;  
    enum {male, female} gender;  
} Person;
```

Data Types

- Linked Data Structure
 - Linear Data Structure with restriction
 - Queue & Stack
 - Linear Data Structure with no restriction
 - Unsorted linked list
 - Sorted linked list
 - Non-linear Data Structure
 - Binary Tree
 - Graph

Linear Data Structure with restriction

- Queue
 - First-In-First-Out (FIFO) data structure
 - the first element added to the queue will be the first one to be removed (post office, bank etc)

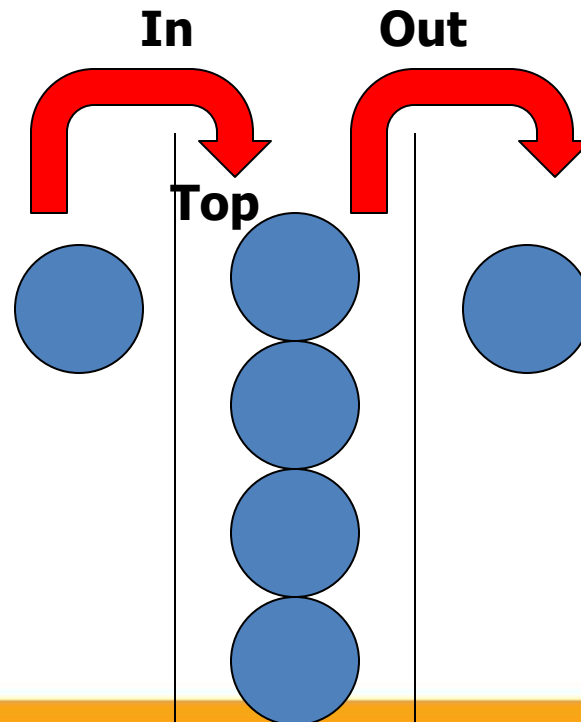


Queue Application



Linear Data Structure with restriction

- Stack
 - Based on the principle of *Last In First Out (LIFO)*
 - Stacks are used extensively at every level of a modern computer system (compiler etc.)

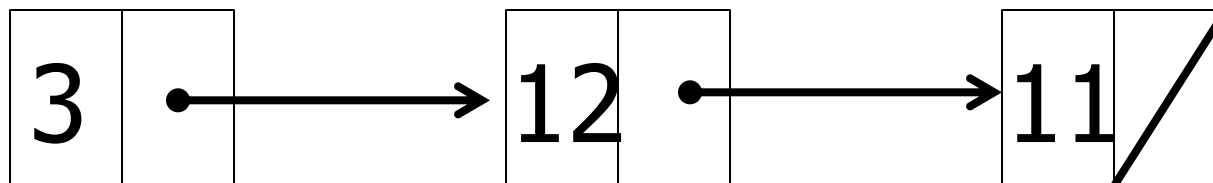


Stack Application



Linear Data Structure with no restriction

- Linked list consists of:
 - a sequence of nodes,
 - data fields
 - one or two links or references pointing to the next and/or previous nodes



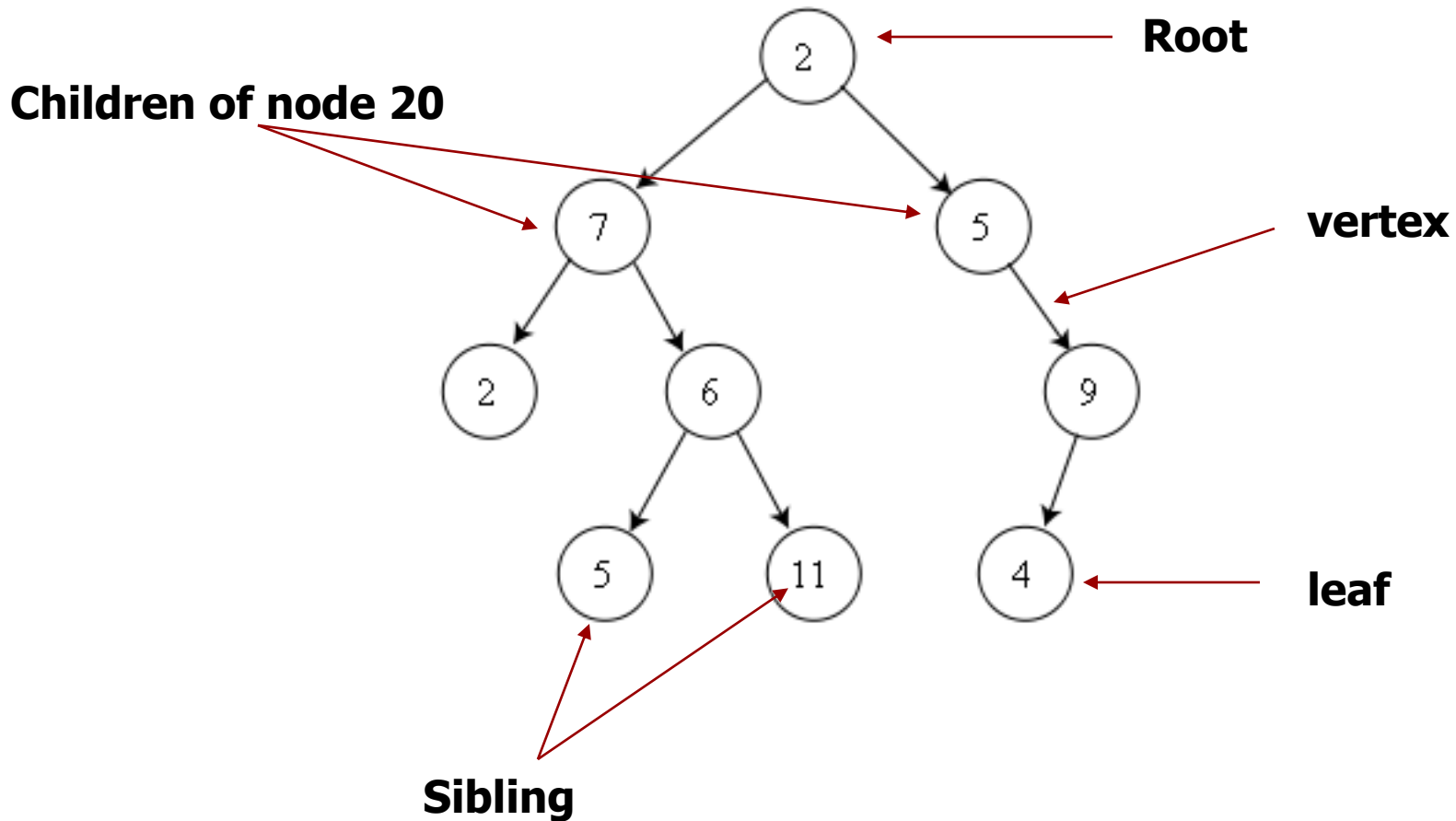
Linear Data Structure with no restriction

- Sorted linked list
 - Data stored in ascending or descending order with no duplicates
 - Insertion at front, middle or rear of the list
 - Deletion will not affect the ascending / descending order of the list
- Unsorted linked list
 - A linked list with no ordering

Non-linear Data Structure

- Tree
 - A data structure based on a tree structure
 - A **tree structure** is a way of representing the hierarchical nature of a structure in a graphical form
 - a **binary tree** is a tree data structure in which each node has at most two children
 - Used for searching big amount of data

Tree



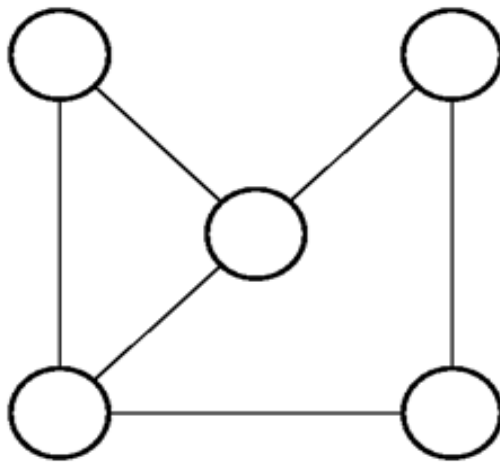
Graph

- A graph consists of a set of **vertices**, and a **set of edges**, such that each edge is a connection between a pair of vertices.
- Some applications require visiting every vertex in the graph exactly once.

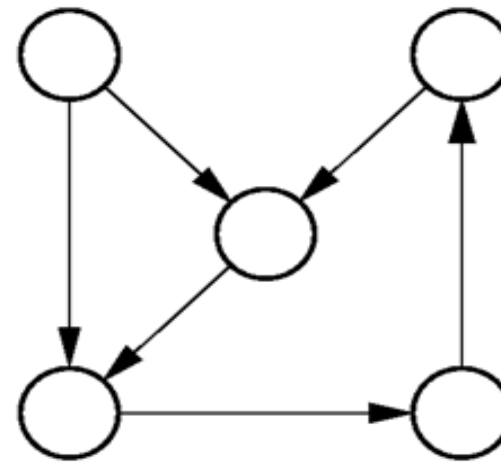
Graph

- The application may require that vertices be visited in some special order based on **graph topology**.
- Examples:
 - Artificial Intelligence Search (Breadth-first search, depth first search)
 - Shortest paths problems
 - Web sites containing a link to and from other websites.
 - Graph that represent courses and the pre-requisites.

Graph Example



Directed



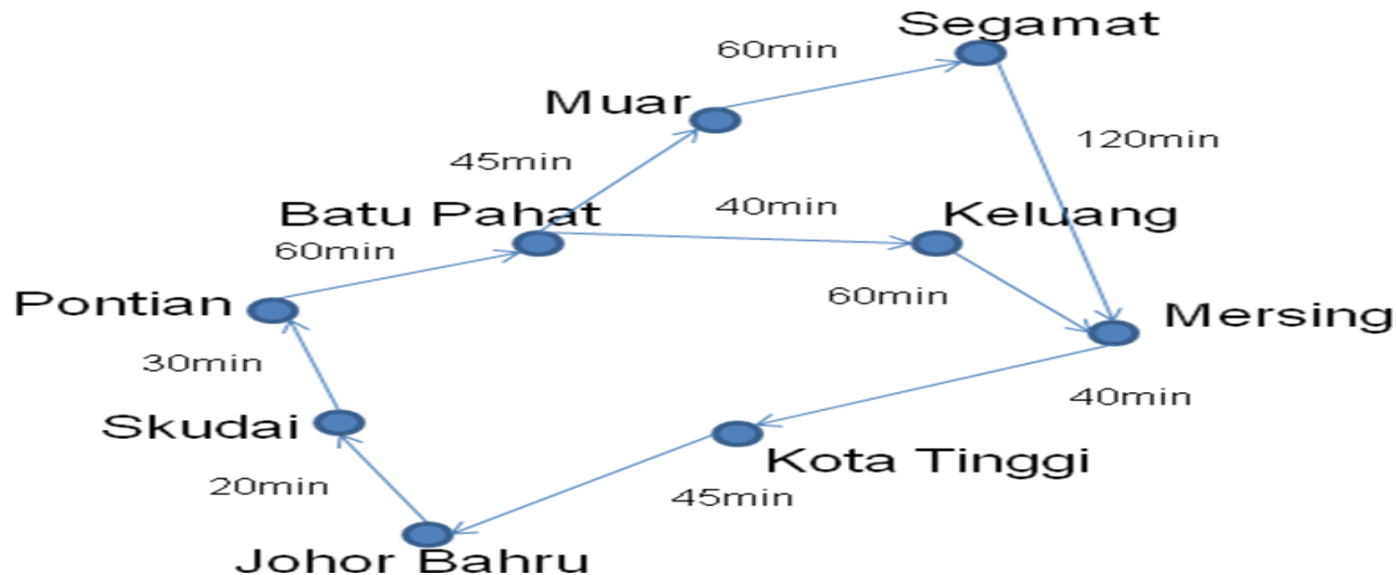
Undirected graph

Network

- Network is a **directed graph**.
- Can be used to represent a route.
- Example :
 - A route for an airline.
 - A route for delivery vehicles.

Network Example

- Weighted network that represents a route for a delivery truck. The route shows all cities in Johor for the truck to deliver items and the time taken for a journey from one city to another.



Conclusion

In this class you have learned about:

- Problem solving is the entire process of taking the **statement of a problem** and **develop a computer program** to solve problems.
- Algorithm is **step-by-step** recipe for performing a task **operate on a collection of data**
- Data structure is a way of **storing and organizing** data in a computer, it allows **efficient algorithm** to be used
- The knowledge given is to ensure that you are able to provide good solution to problem solving

References

- Frank M. Carano, Janet J Prichard. *“Data Abstraction and problem solving with C++” Walls and Mirrors*. 5th edition (2007). Addison Wesley.
- Nor Bahiah et al. *“Struktur data & algoritma menggunakan C++”*. Penerbit UTM. 2005.