Introduction to Data Structures and Algorithms

Le Quy Loc

from Da Nang university of technology

Problem Solving

- Problem solving =
  - Understand problem
  - **Design solution**
  - Implementation

Understand Problem

- Problem description
  - Input data
  - Output data

Algorithm - 1

- Algorithm is a sequence of steps to solve a problem in information technology major

Algorithm - 2

- The characteristics of algorithm
  - Finiteness
  - Uniqueness
  - Generality
  - **Effectiveness**
    - **Execution time (complexity)**
    - Consumed memory

Algorithm - 3
Pseudo Code

- A draft of algorithm
  - Written in natural language
  - And easy to be converted into programming languages
- Ex: sum of integers from 1 to n *(while style)*
- Most benefit of algorithm
  - Help programmers focus on designing solution for a problem
  - Don’t care about programming languages

Algorithm - 4

- Algorithm doesn’t depend on
  - Programming languages
  - Computer hardware

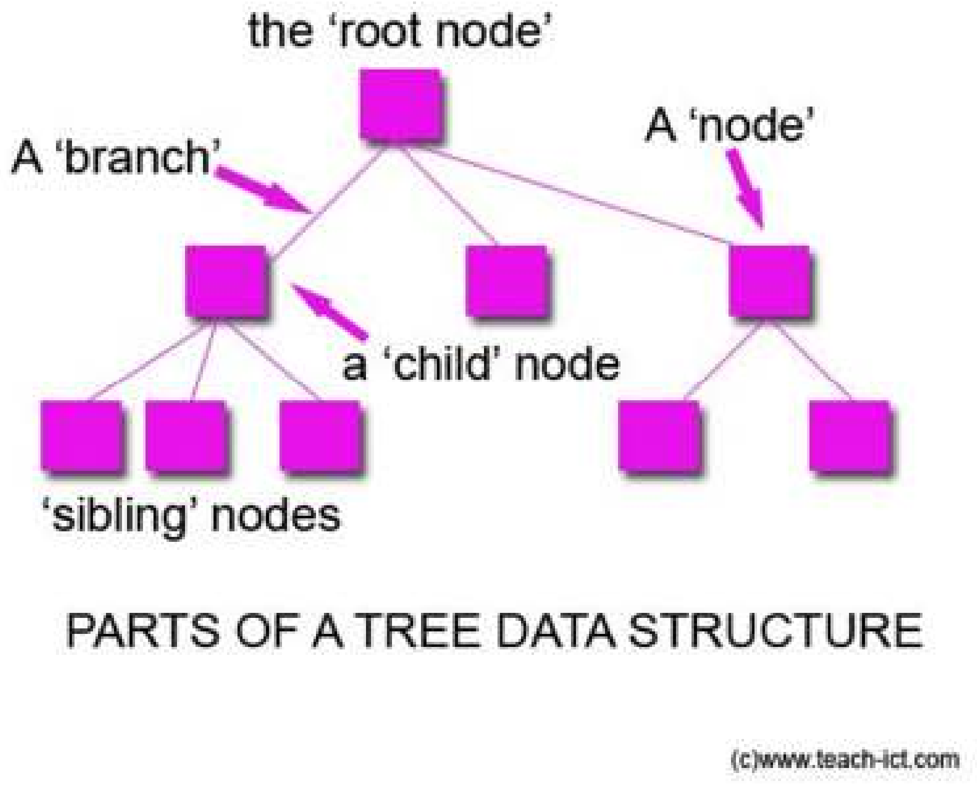
Homework

- Algorithm to sort an array increasingly

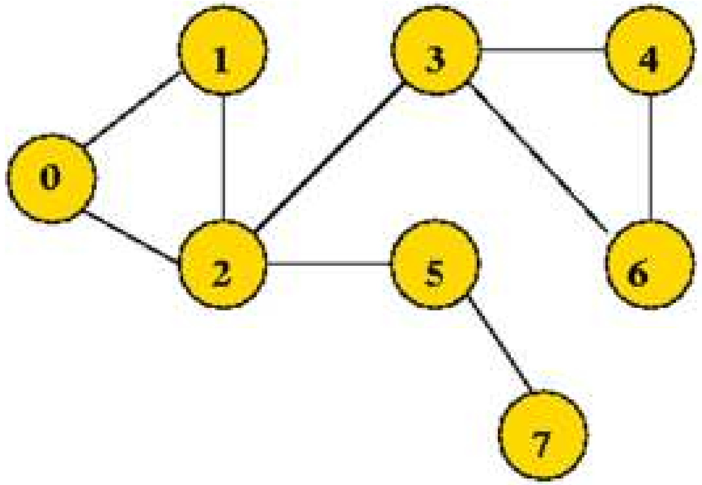
The Interaction between Algorithm and Data

- Algorithm always interact with data
  - The way of organizing data affects substaintialy to the effectiveness of algorithm

The Interaction between Algorithm and Data

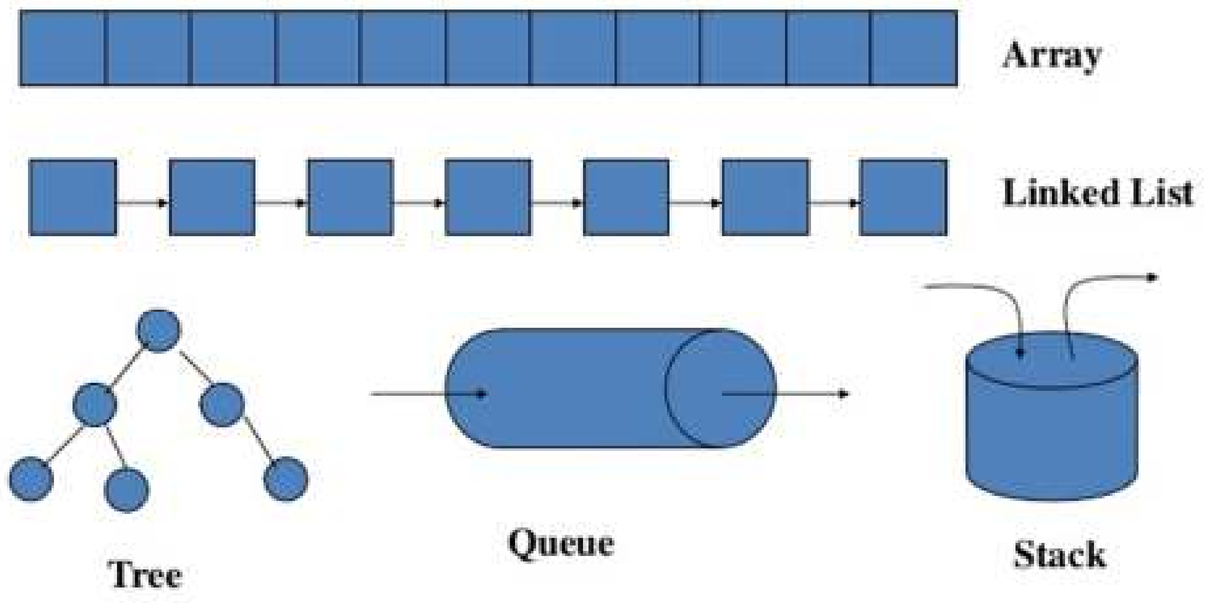


The Interaction between Algorithm and Data



The Interaction between Algorithm and Data

Types of data structures



There are many, but we named a few. We'll learn these data structures in great detail!

The Interaction between Algorithm and Data

- Solution design = Algorithm + Data Organization

Abstraction - 1

- Many problems can be modeled with the same rule
  - Ex: wear and put off many t-shirts, put and remove batteries from a flashlight
  - Put and remove an element at the end of the set

Abstraction - 2

- We can design a model to solve a class of problems with the same rule

Abstraction – 2
Data Abstraction

- A model includes
  - Data organization
  - Operations

→ Data Abstraction

Modularity - 1

- A solution for a problem usually includes some classes
- The classes interact each other via methods

Modularity - 2

- Why do we have to modularitize?
  - Improve the quality of software development
  - Reuse the solution

Abstract Data Type

- Simple data types
  - Integer, real, character
  - Array, string
- So what is ADT?
  - A packaging solution to solve a class of problem
- Ex
  - Stack, Queue, List, Tree, Graph

Data Structures

- Data structures: a way to build an ADT
- A data structure includes
  - Data
  - Operations

Objectives

- Introduce popular data structures & algorithms
  - Design, implementation and application
- Select a right data structure or customize a data structure to solve a problem
- Analyze the effectiveness of algorithm

Exercises

- Look for real examples with the rules
  - Stack: push and pop an element at the end of the sequence
  - Queue: add an element at the end of the sequence, and remove an element at the front of the sequence
  - List: add/remove an element at any position in the sequence

Exercises

- List the possible operations of the ADTs
  - Stack
  - Queue
  - List

Evaluation

- Attendance & homework: 20%
- Midterm: 20%
- Final: 60%

C Programming Language Review

- Sort an array increasingly
- Write a function to exchange the values of 2 variables