Spring 2019 (Course Number 009952)

Data Structures and Lab
(Lecture 01: Introduction)

Prof. A. P. Shrestha, Ph.D.
Dept. of Computer Science and Engineering, Sejong University

Today

- Information about this course
- Evaluation, grading and other rubrics
- Brief course overview

Next class

- Array
- Basic operations and sorting in array

Course Information

- Course Number 009952
- Credits: 4
- Hours: 5 hours/weekly

  Tuesday 12:00 noon - 1:30pm(센 B202), and
  6:00pm - 8:00pm(동 415)

  Thursday 12:00 noon - 6:00pm (센 B202)
- Class Type:

  Theory: 3 hours(90 minutes each)

  Lab: 2 hours

  The slides and lab sheets/assignments will be available in Blackboard

Teaching Assistant

Mr. Abdukhakimov Farshed

Email: farshed888@gmail.com

- Your queries related to lab/assignments can be sent to me or TA.
- Your lab/assignments will be checked by me and TA.

Weekly Lecture Plan

Week	Contents
Week 1	Course Information, Array
Week 2-6	Linked Lists, Stacks, Queue and their implementation and applications
Week 7	Algorithm analysis and revision for midterm
Week 8	Mid Term
Week 9-14	Tree, Self-balancing Tree, Priority Queue, Heap, Hash Table, Graph
Week 15	Team Project Presentation
Week 16	Final Exam

*Quiz 1 will be between week 2-6
*Quiz 2 will be between week 9-14
*Short notice will be given prior to quiz. Please study every week!!

Pre-requisite

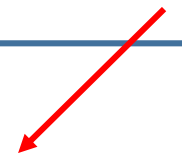
- Students should have proper programming skills and be able to use
  - Structure and Class
  - Pointers
  - Dynamic memory allocation
  - Templates (is useful in C++)

- Programming language
  - C++ language
  - C language (is fine)
  - Any compiler is fine. Algorithm is important

- Tips
  - Always write comments in your code

Evaluation

중간고사(%)	기말고사(%)	수시평가및과제(%)	출석(%)
30	30	30	10



Attendance (10%)	**Mid-term (30%)**	**Final-Exam (30%)**	**Others (30%)**
1 or no absence: 10	Objective questions	Objective questions	Lab/Assignments: 15
Up to 3 absence: 9	Short answers	Short answers	Quiz 1: 3
Up to 5 absence: 8	Programming/ Debugging questions	Programming/ Debugging questions	Quiz 2: 4
More than 5 absence: 6Abs:7,7abs:6, 8abs: 5 and so on			(Team) Project: 8
More than 12 absence-FA			

- Total Attendance: 2×16 +16-3 = 48-3=45
- Quiz 1 (before midterm) and quiz 2 (after midterm) will have only objective questions

Attendance

Department Rule!!

Present	Late	Absent
After 5 mins lect. start	5-15 mins after lect. start	After 15mins

Procedure

- Install Ucheck Plus app in your smartphone and check-in accordingly

Note

- If there is any issue during check-in, inform Professor before 15 mins.
- Delay in reporting will result in absence
- Week 1 attendance will not be counted during grading

Grading

*m: mean
*d: deviation

- Grading is **relative!**
- **Criteria 1:** Top 25% (approx.) can get A+ or A (university recommendation)
- **Criteria 2:** At least 50% should be scored in assignment/lab and project work to get B or above.

Grades	Criteria 3	Criteria 4
A+	> m+1.5d	85 % or above
A	m+1.0d ~ m+1.5d	70% or above
B+	m+0.5d ~ m+1.0d	NA
B	m ~ m+0.5d	NA
C+	m-0.5d ~ m	NA
C	m-1.0d ~ m-0.5d	NA
D+	m-1.5d ~ m-1.0d	NA
D	m-2.0d ~ m-1.5d	30% or above
F	NA	Less than 30%

All four criteria should be fulfilled!!

Course Project

- Open end Project
  - You can select your own topic related to course
  - Discuss your project topic with Professor for approval
  - After approval, you can start your project
  - You have to present your project on week 15
  - Maximum 2 team members (If you want, you can do alone!!)

Assignments & Lab

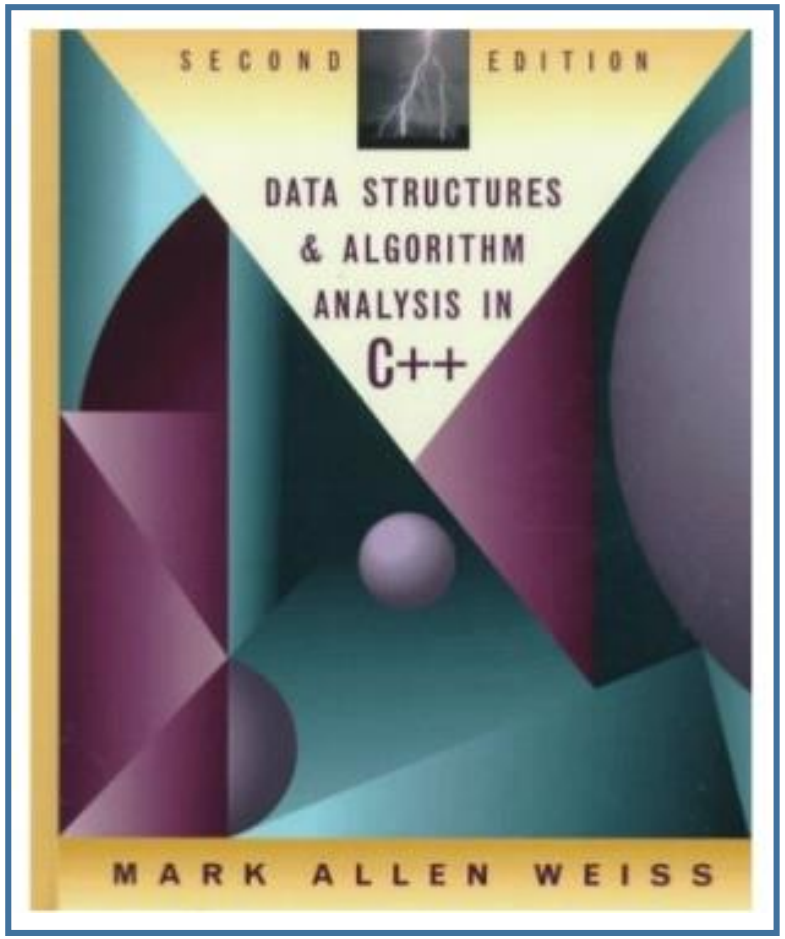
- Lab sheets will have multiple questions
- Points will be indicated in each question
- Assignments will also be included in lab sheets

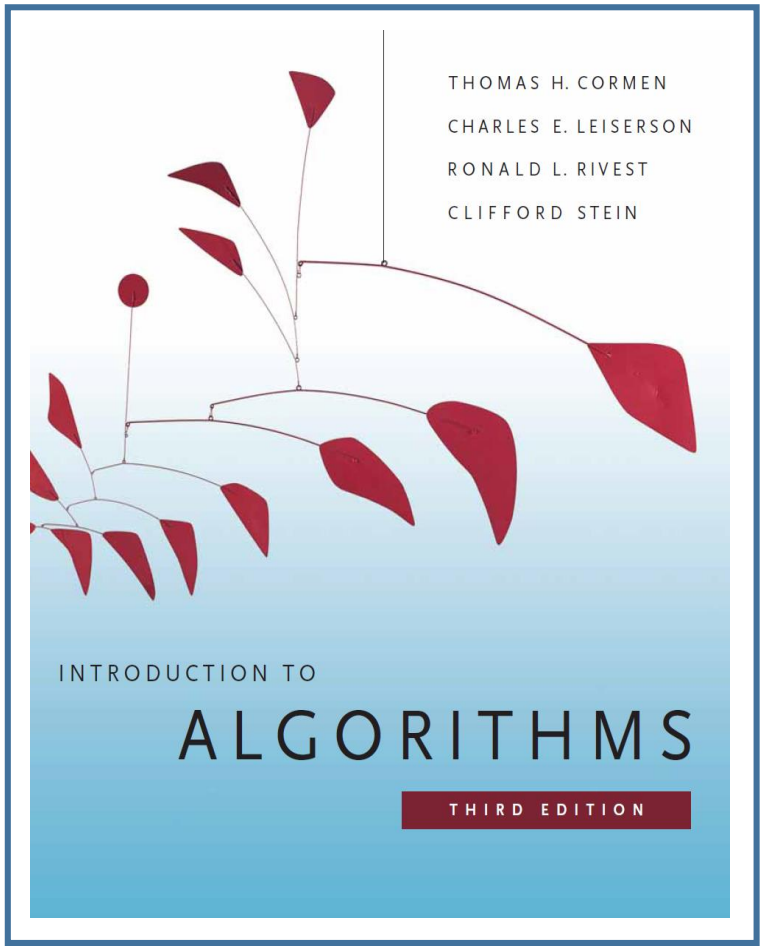
- **Assignment/Lab Rubrics**

	0 score	50% score	100% score
Completion	No attempt	Partial attempt	Fully completed
Correctness	Does not address question	Partially correct	Fully correct
Comments	No comments	Only basic comments	Comments on critical steps

Deadline (9 days)	After 14 days : 0.0 ×Total	Within 9~14 days: $^1/_2$×Total	Within 9 days: Total

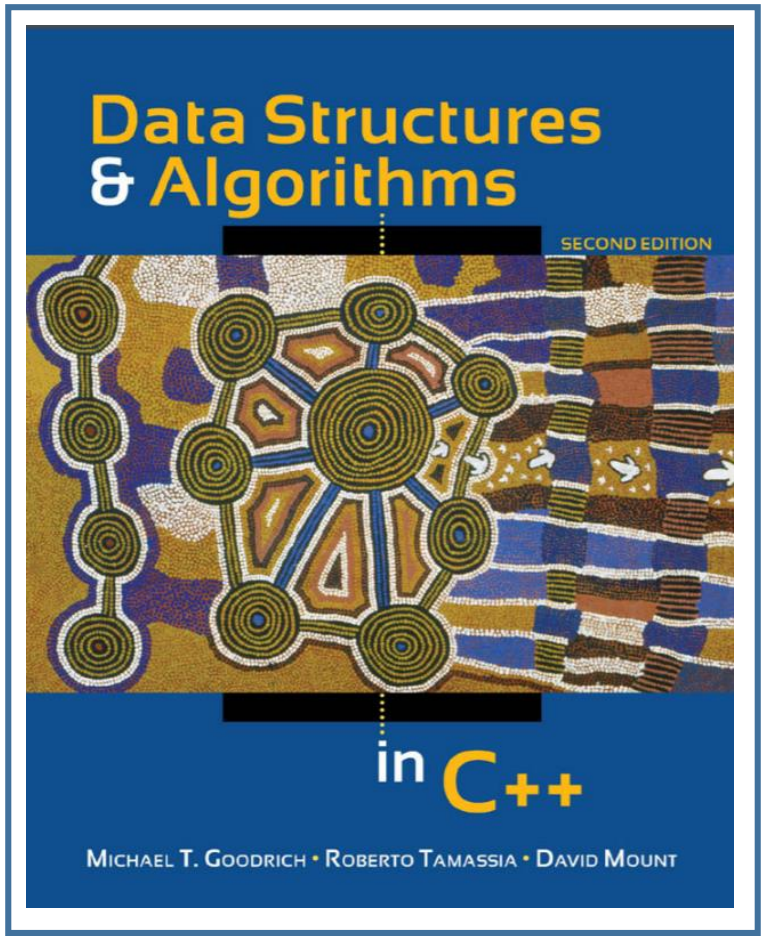
Reference Books

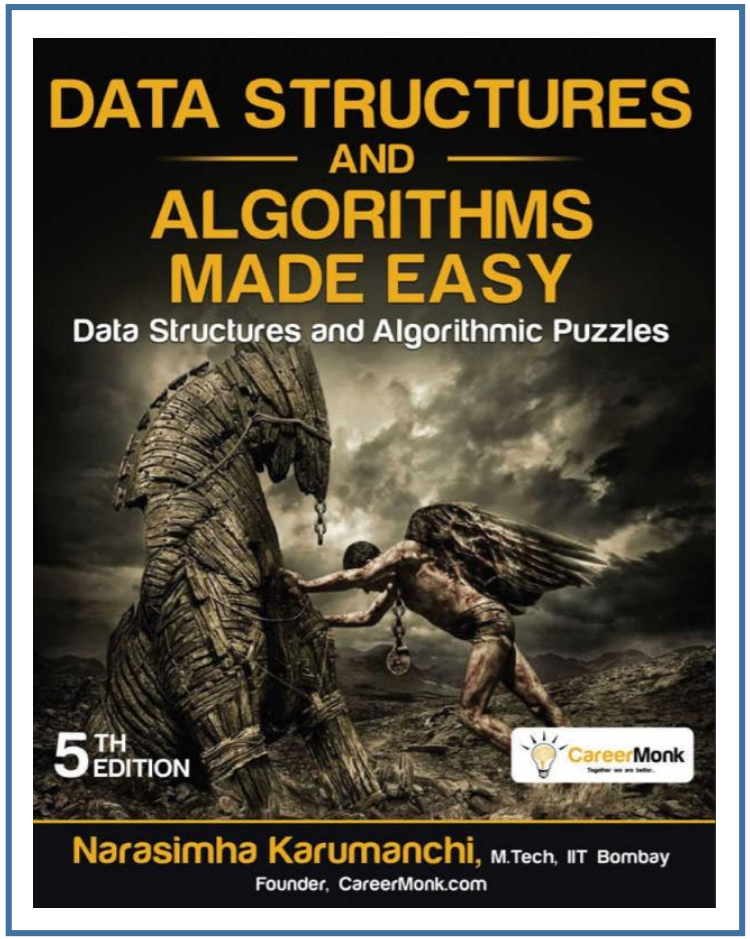




Note: Lecture materials are not based on one particular book, but several books and online materials

Reference Books





Note: Lecture materials are not based on one particular book, but several books and online materials

Value vs. Success

"Try not to become a woman/man of success. Rather become a woman/man of value."

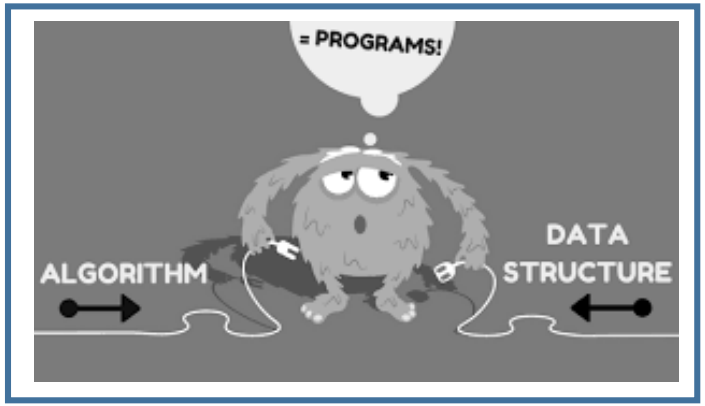
- Do it yourself!!
- No plagiarisms
- No copying assignment (discussion is fine).
- No cheating in exam

Contact

- Email (recommended)
  - anishpshrestha@sejong.ac.kr
  - Please mention your name and course name while sending email
- Office Phone
  - 2-6935-2445
- Mobile Phone
  - Available in blackboard. Do not hesitate to call me in mobile if it is urgent.
- Office Time
  - Preferred: Wed 2:00-3:30 pm, Fri 2:00-3:30 pm
  - Please make a prior appointment through e-mail (recommended).
- Office
  - Room# 431 센

Brief Course Overview

1.1 Whats and Whys



- **Algorithm**
  - step-by-step instructions for solving problem
- **Program**
  - an implementation of the algorithm in some programming language
- **Data Structure**
  - Efficient organization and storage of data needed to solve the problem
  - supports certain operations, each with a:
    - Meaning: what does the operation do/return
    - Performance: how efficient is the operation

1.1 Whats and Whys

- Study of data structure includes the following three steps:

1. Logical or mathematical description of the structure

2. Implementation of the structure on a computer

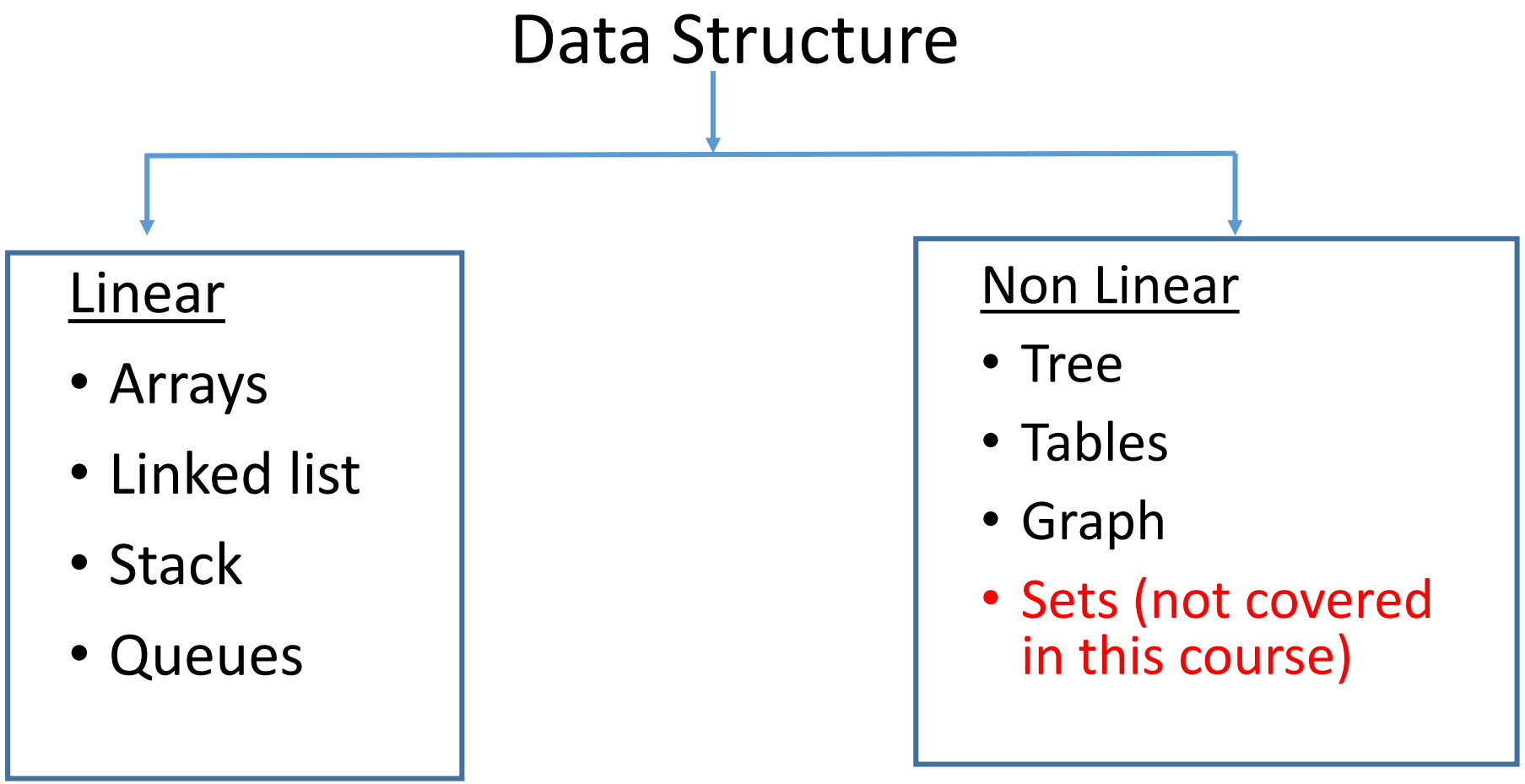
3. Quantitative analysis of the structure
   - Amount of memory needed to store the structure, and
   - Time required to process the structure

1.2 Data Types

- A classification of data which tells the compiler or interpreter how the programmer intends to use the data

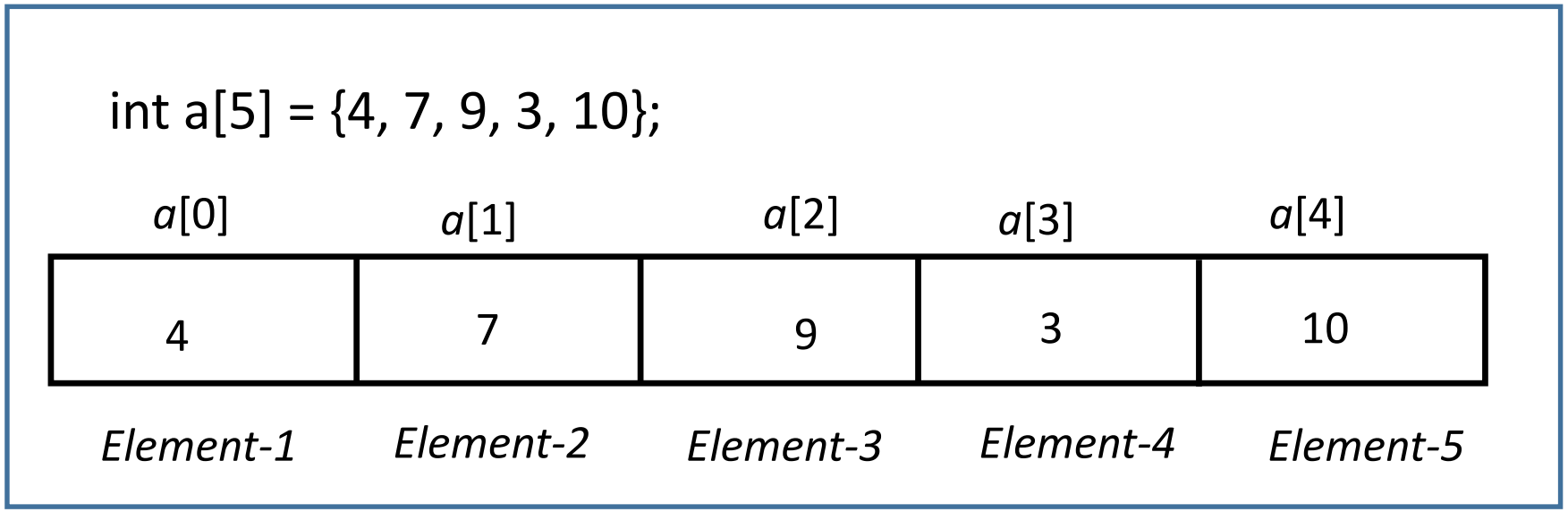
- Typical Built-in Data Structures
  - Character
  - Integer
  - Float
  - Double
  - Pointer

1.3 Classification of Data Structure



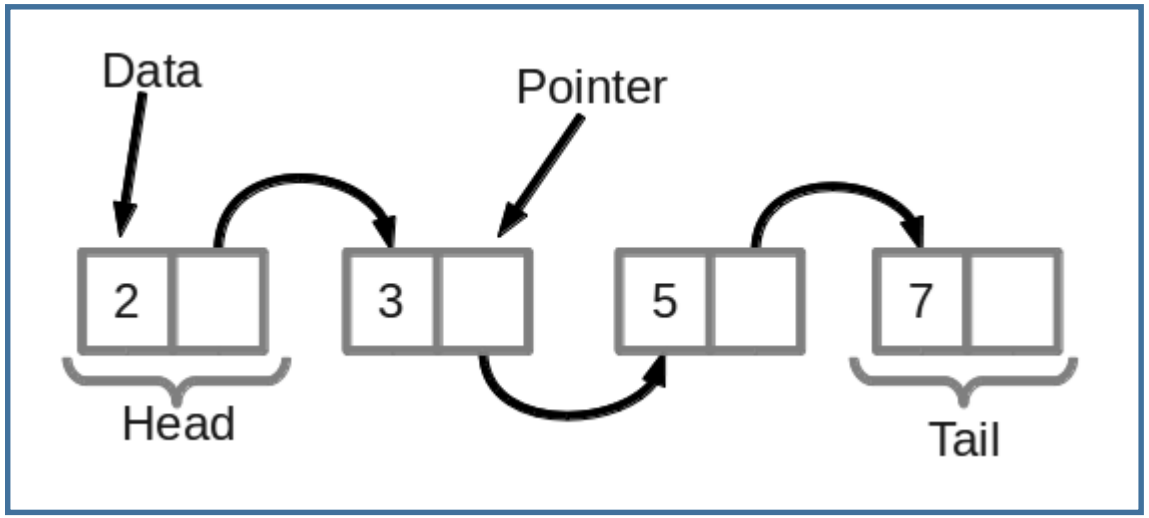
1.4.1 Array

- Arrays can store a fixed-size sequential collection of elements of the same type.



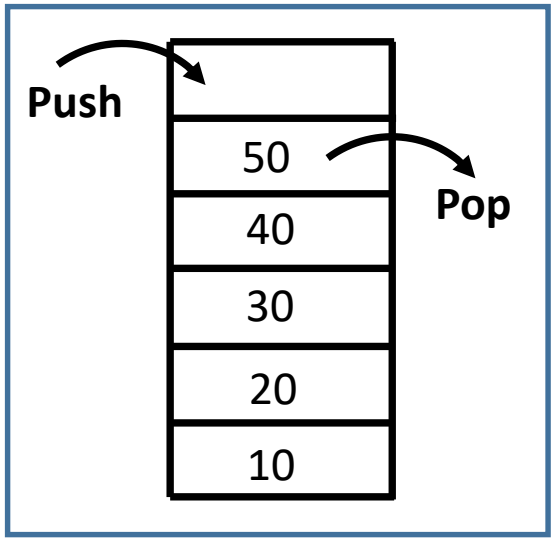
1.4.2 Linked List

- Linked list is similar to arrays which store a linear data structure.
- However, linked list elements are not stored at contiguous location; the elements are linked using pointers.
- Each node in a list consists of at least two parts:
  i) data, and ii) pointer to the next node



1.4.3 Stacks

- Stack is a linear data structure
- It follows a particular order in which the operations are performed.
- The order may be LIFO(Last In First Out) or FILO(First In Last Out).



- **Push**

  Adds an item in the stack.

- **Pop**

  Removes an item from the stack.

  The items are popped in the reversed order in which they are pushed.

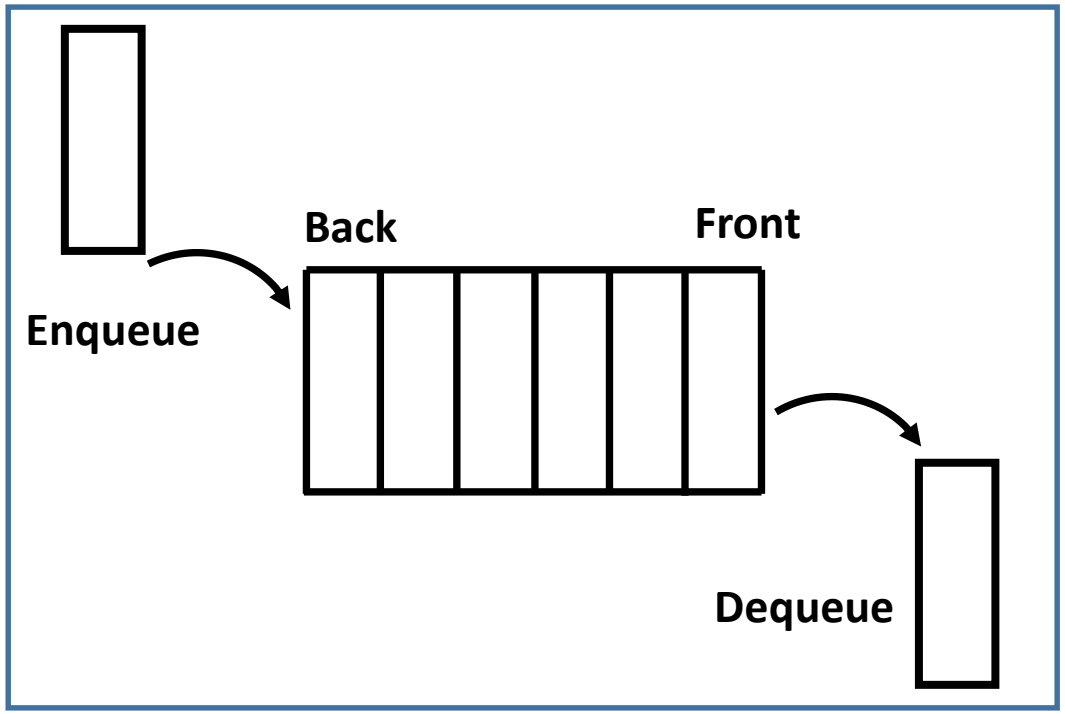
1.4.4 Queue

- Queue is a linear data structure
- It follows a particular order in which the operations are performed.
- The order may be FIFO(First In First Out)
- **Enqueue**

  - Adds an item in the queue.
- **Dequeue**

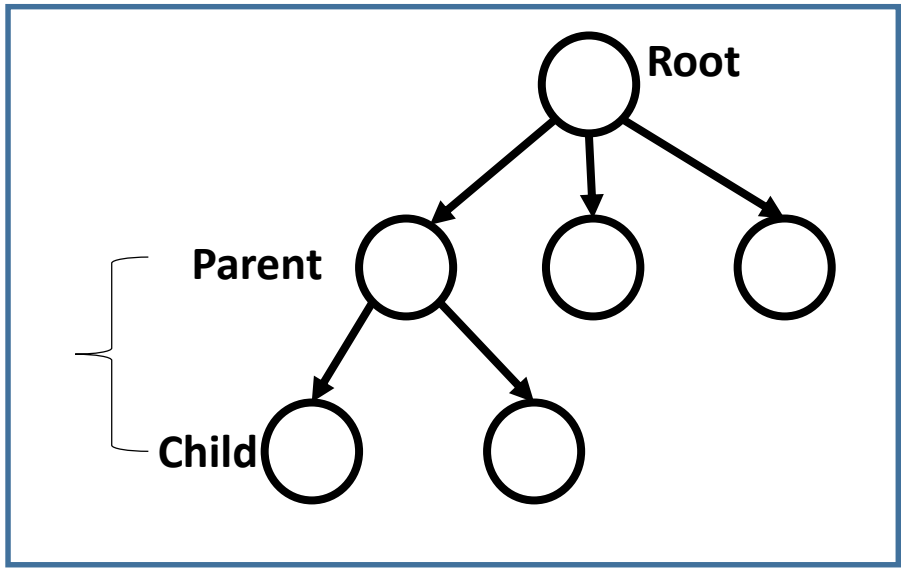
  - Removes an item from the queue.

  - The items are popped in the same order in which they are pushed.



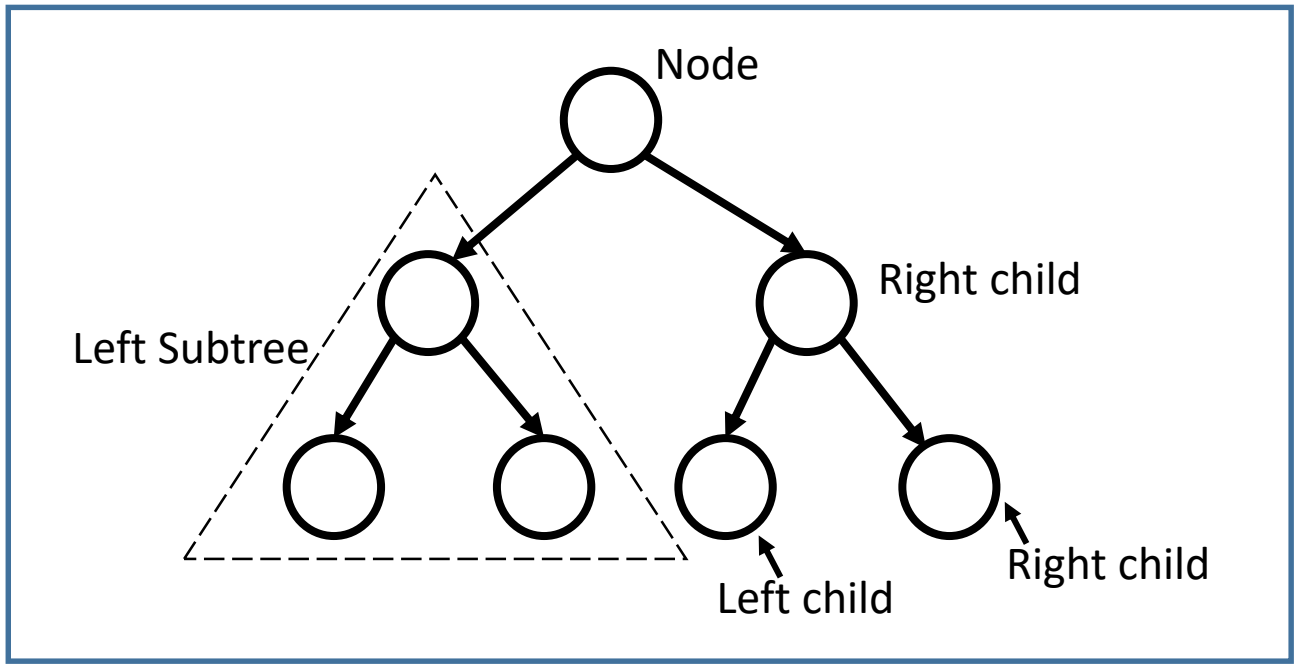
1.4.5 Tree

- Trees are hierarchical data structures
- The topmost node is called root of the tree.
- The elements that are directly under an element are called its children.
- The element directly above something is called its parent.



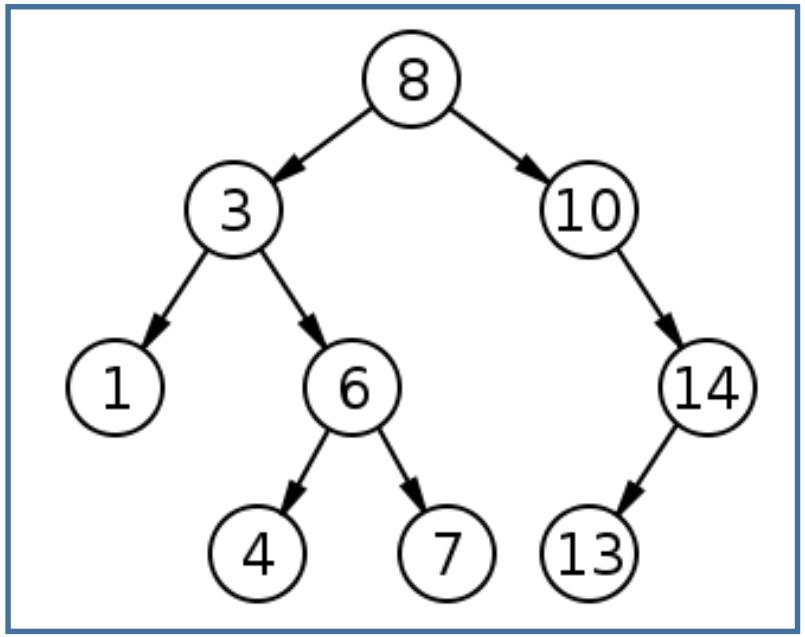
1.4.6 Binary Tree

- A tree whose elements have at most 2 children is called a binary tree.



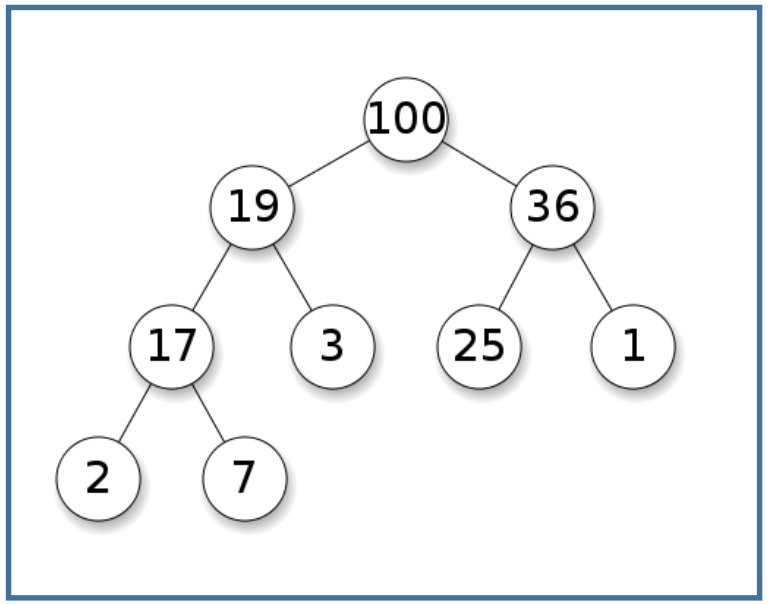
1.4.7 Binary Search Tree

- Binary Search Tree has the following properties:
  - The left subtree of a node contains only nodes with keys lesser than the node's key.
  - The right subtree of a node contains only nodes with keys greater than the node's key.
  - The left and right subtree each must also be a binary search tree.



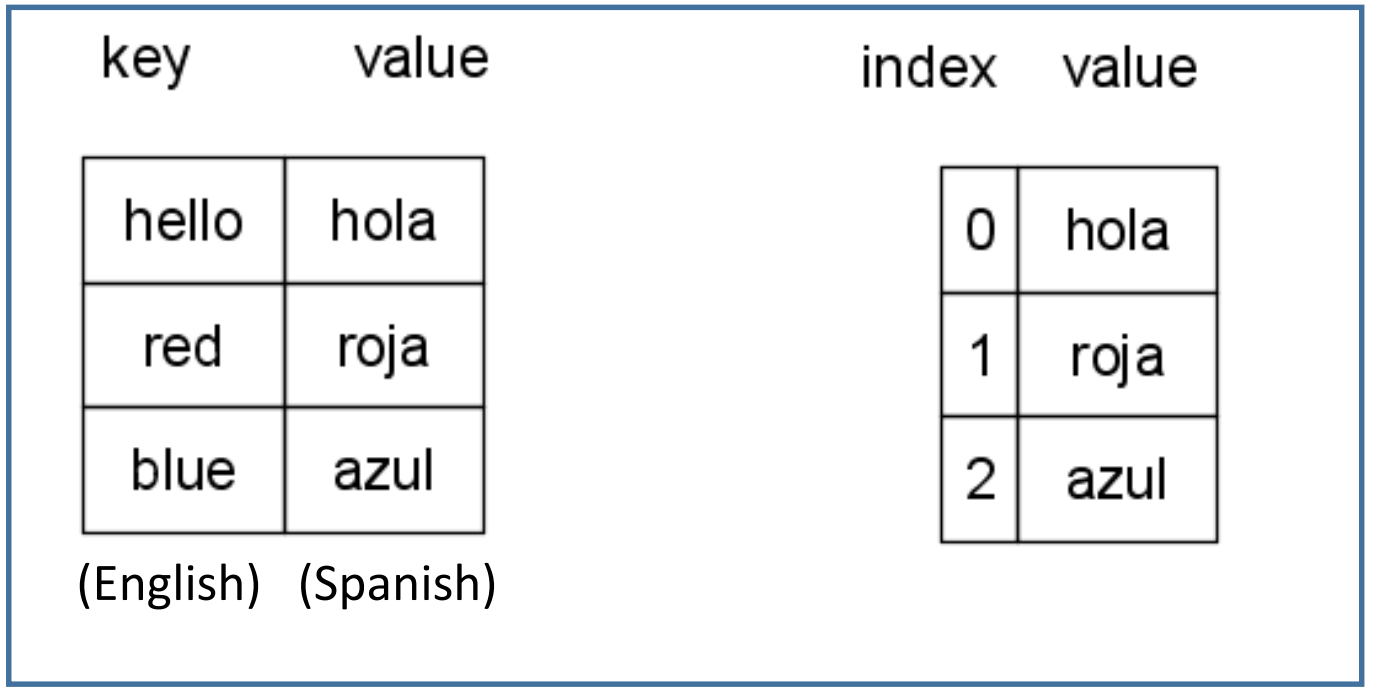
1.4.8 Binary Heap

- A Binary Heap is either Min Heap or Max Heap.
- In a Max Binary Heap, the key at root must be maximum among all keys present in Binary Heap.
- The same property must be recursively true for all nodes in Binary Tree.
- Min Binary Heap is similar to Max Heap.



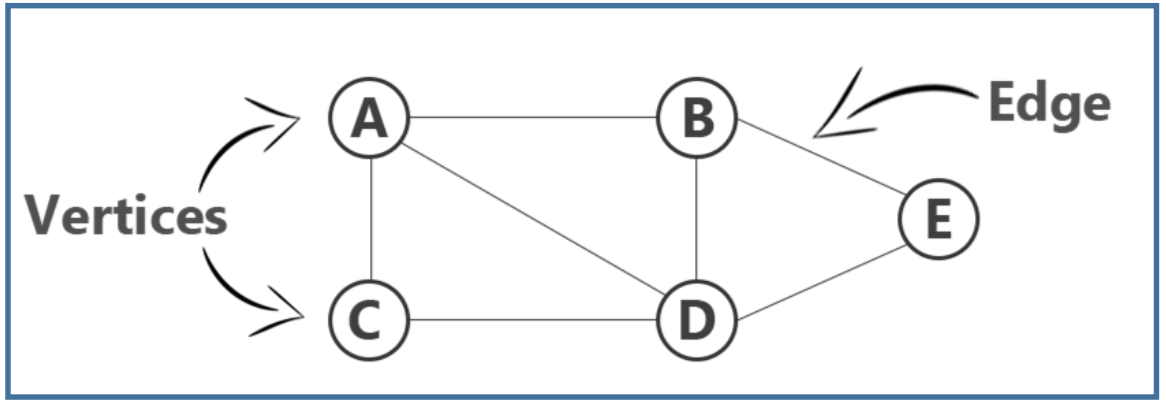
1.4.9 Hash Table

- A data structure that can map keys to values
- A hash table uses a hash function to compute an *index*



1.4.10 Graph

- Graph is a data structure that consists of following two components
  - A finite set of vertices also called as nodes.
  - A finite set of ordered pair of the form (u, v) called as edge.



This graph G can be defined as G = (V , E)
where V = {A,B,C,D,E} and
E = {(A,B),(A,C)(A,D),(B,D),(C,D),(B,E),(E,D)}.

1.5.1 Operations

- **Traversing**
  - Accessing each and every element present in a data structure like an array or a linked list at least once.
  - Also known as visiting the record.
  - An example if you want to print an array

- **Searching**
  - Finding the location of the record with a given key value, or finding the locations of all record which satisfy one of more conditions

1.5.2 Major Operations

- **Sorting**

  - sort items in certain order

  - Example: i) ascending or descending order in case of numerical data
    ii) in dictionary order in case of alphanumeric data.

- **Merge (/Sort)**

  - It is used to combine the data items of two sorted files into single file in the sorted form.

1.6.2 Major Operations

- **Inserting**
  - Adding one or more new data elements to the structure
  - Based on the requirement, new element can be added at the beginning, end or any position

- **Deleting**
  - Removing an existing element from the array
  - After removal, we may have to reorganize all elements in the structure

Q & A ?