

# CSC 315 Data Structure and Algorithm Analysis

Eliyas Girma(MSc)

# Contents

- Chapter 1 -Introduction to Data Structures
  - Abstract data types (ADT)
  - Abstraction
- Chapter 2 -Algorithm and Algorithm Analysis
  - Properties of algorithms
  - Analysis of algorithms
- Chapter 3-Simple sorting and searching Algorithms
  - Sorting
  - Searching
- Chapter 4 – Data structures
  - Review on pointers and dynamic memory allocation
  - Linear and Non-linear ADT
- Chapter 5 –Advanced Sorting and Searching
  - Sorting
  - Advanced and searching

- Assessment
  - Lecture, Lab and Assignemet
  - Mid (20%), Final(40%), project(20%),
  - Assignment 1 (10%)
  - Assignment 2 (10%)
- Text Book
  - Mark Allen weiss:Data structure and problem solving using c++,USA, Addison wesley 1999

- All course Materials and Text books are found at

[https://github.com/elicho99/Data\\_structure](https://github.com/elicho99/Data_structure)

# Chapter 1 -Introduction to Data Structures

# Data structure

- Ways of storing data  
Our Main Focus: storage in main memory(RAM)
- Data structure refers to the way data is stored, and organized in computer systems
- It is the method of organizing large amount of data.
  - Linear data structures  
Array, Linked list , Stack, Queue
  - Non linear data structures  
Trees, Graphs, Hash Maps

- Common Operations of data structures
  - Traversing and searching
  - copying and cloning
  - insertion and deletion
  - Merging and sorting
- Choice of data structure depends on two things
  - It must be rich enough in structure to mirror the actual relationship of data in real world
  - It must be simple enough to process for manipulation of data

# Why Study Data Structures?

- To gain literacy on basic data structures and algorithms
- To become a better programmer and have better algorithmic problem solving skills.
- It might help you ace technical interviews!
  - Google, Microsoft, Amazon Facebook, ...
  - Numerous other international and a few local software companies



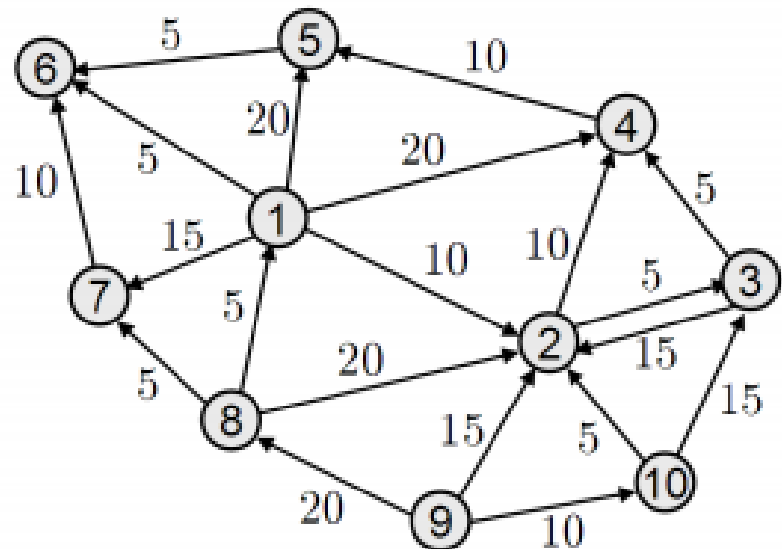
# Example1: Queues

- As the name implies: a first-in, first-out(FIFO) list
- Application examples:
  - Queuing requests in large-scale systems (e.g. web-servers, SMS gateways, call center ...)
  - Event scheduling and simulation



# Example 2: Graphs

- Nodes connected using edges
- Can be used to model numerous situations such as:
  - Social Networks(facebook, twitter, ...)
  - Internet Routing
  - Web Search
- Nodes: computers, switches, routers and other active devices
- Edges: Connection between these active devices
- The basis for routing algorithms



# Data types

- Are data abstractions provided by the programming language
- Example: integer data type
  - You don't have to worry about the representation of integers in memory (two's complement, sign and magnitude ...)
  - You also don't have worry about the memory addressing system
  - Likewise, you don't have to know how the arithmetic operations are performed in the CPU

# What Exactly is Data Abstraction?

- Way of expressing data in with a representation similar inform to its meaning
- Implementation Details are hidden from the user.
- Enables us to limit our focus to the problem domain
  - Reduced complexity
- Abstraction is the process of leaving details and providing general specification while solving a problem.

# Abstract Data Types(ADTs)

- Enables us to create data abstractions that represent more complex data structures
- Set of data values and associated operations independent of any particular implementation.
- OOP is a great way to create these abstractions

