FEE 231: Computer Science III

Data Structures and Algorithms

Introduction

- Data Structures and Algorithms
- Prerequisite: FEE 131 and FEE 132
  - Knowledge of a high-level programming language: C/C++
- Review and strengthen your programming: tutorials, etc.
  - Reference text, Chapter 1 & 2
  - Algorithms are language-independent
  - Implementation requires knowledge of a language

Introduction

- Theory Classes (2 hours/week)
- Lab work (2 hours/week)
- Online Content:
  - E-Learning platform (https://learning.uonbi.ac.ke)
    - Course name: FEE231; enrollment key: FEE2312018)
  - Sharif Judge
- Assessment:
  - Quizzes & Exercises (theory and programming) [20%]
    - Do your own work!!
    - Struggle until you get it right!!
  - CAT(s) [10%]
  - Exam [70%]

Course Objectives

- At the end of the course the student should be able to **develop and code complex data structures** required for a variety of electronic and data management systems e.g., memory management, graphic drivers, computer-aided design (CAD), data base systems, etc.

- The student will be expected to learn advanced programming techniques required for implementation of **dynamic data structure algorithms** using a structured high-level programming language.

Introduction

- **Course Outline: get it from the eLearning Portal**

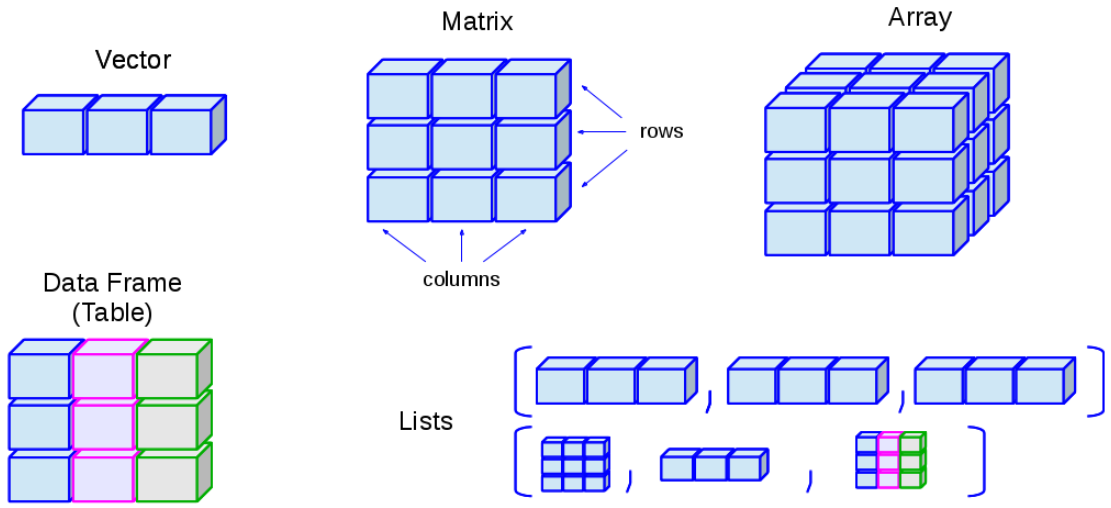
- **Reference Text**
  - Michael Goodrich et al, *“Data Structures and Algorithms in C++”*

Algorithm

- Informally, an algorithm is any well-defined **computational procedure** that takes some value, or set of values, as **input** and produces some value, or set of values, as **output**.
- An algorithm is thus a **finite** sequence of computational steps that transform the input into the output
  - An algorithm can be specified in English, as a computer program, or even as a hardware design.
  - Must provide a precise description of the computational procedure to be followed

Data Structures

- A ***data structure*** is a way to store and organize data in order to facilitate access and modifications.
- No single data structure works well for all purposes, and so it is important to know the strengths and limitations of several of them.



Applications of Algorithms & Data Structures

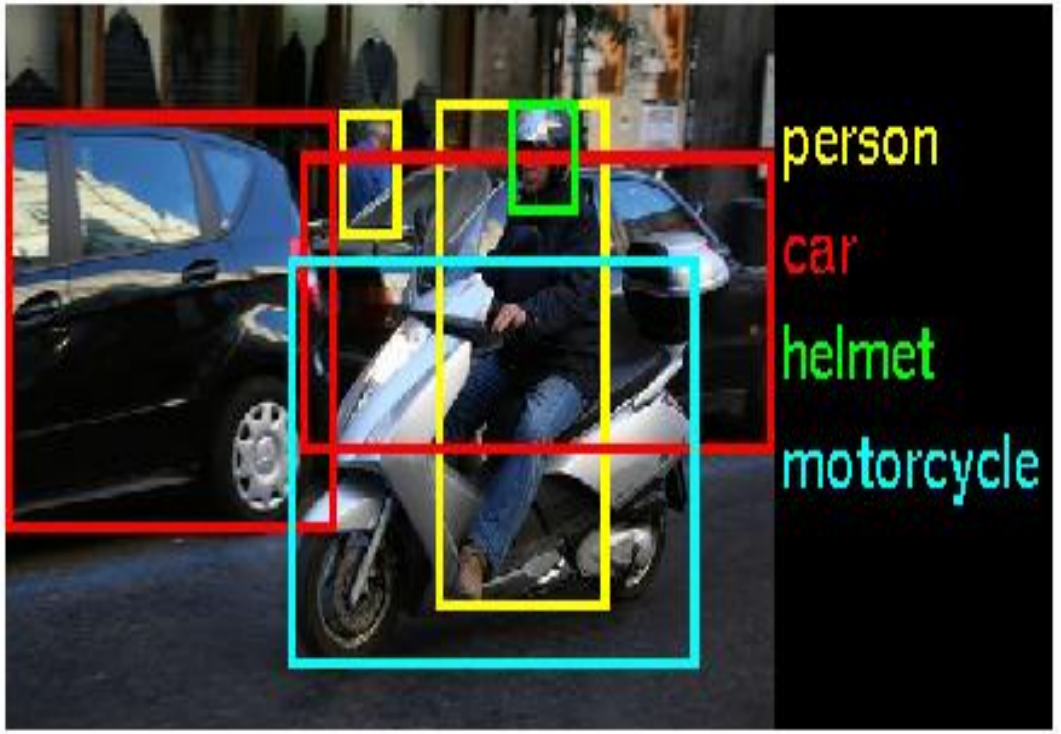
- Programming in Engineering
  - Computer Control
  - Simulation, etc
- Digital Signal Processing
  - Powerful Computers; A-D converters
- Telecommunications: GSM, Satellite technology, WiFi, etc
  - Linear Predictive Coding
  - Information Theory
- Heavy Power
- Embedded Systems
  - Efficient programs to control hardware
- Internet: manage and manipulate large volumes of data
- Ecommerce:
  - Encryption, etc
  - BitCoin

Applications of Algorithms & Data Structures

- Google:
  - Page Rank => searching
  - Google Maps: shortest path, etc
- Machine Learning and Artificial Intelligence
  - Deep learning: Google, Amazon, Facebook,…
    - Amazing products to their users
    - Google bought DeepMind in 2014
  - Self-driving cars, Face Recognition, Speech and Language Processing
  - Chess games, GO: AlphaGo vs Fan-Hui (5:0); vs Lee Sedol (4:1)
  - IBM's Watson: recently used in Wimbledon, among others
  - Has been compared to the **Industrial Revolution**
  - **Andrew Ng: AI is the new electricity!**

Applications of Algorithms & Data Structures

- Convolutional Neural Networks
  - ImageNet Classification Challenge:
  - 72% in 2010 (Humans 95%)
  - 85% in 2012
    (Geof Hinton, Deep Learning)
  - 96% in 2015
- Applications in Computer Vision
  - Object detection and classification
  - Image Captioning
  - Visual Question Answering
  - Paragraph Generation



Applications of Algorithms & Data Structures

- Smart Systems and Automation
  - Building Management Systems (BMS)
  - Smart Homes
  - In Agriculture
  - In Manufacturing

- Most applications have algorithmic content, explicitly or implicitly

Data Structures and Algorithms

- They necessarily come up when dealing with programs to solve problems
- There is a need for good data structures and good algorithms
- Measure of goodness of algorithms:
  - **Efficiency**
  - **Correctness**
- Dealing with big data sets?
- Need for algorithms that not only SOLVE problems but also solve them in the BEST way.

Data Structures and Algorithms

- Computers may be fast, but they are not infinitely fast
- Memory may be inexpensive, but it is not free.
- Computing time and space in memory are bounded resources

- Algorithm knowledge: design better solutions