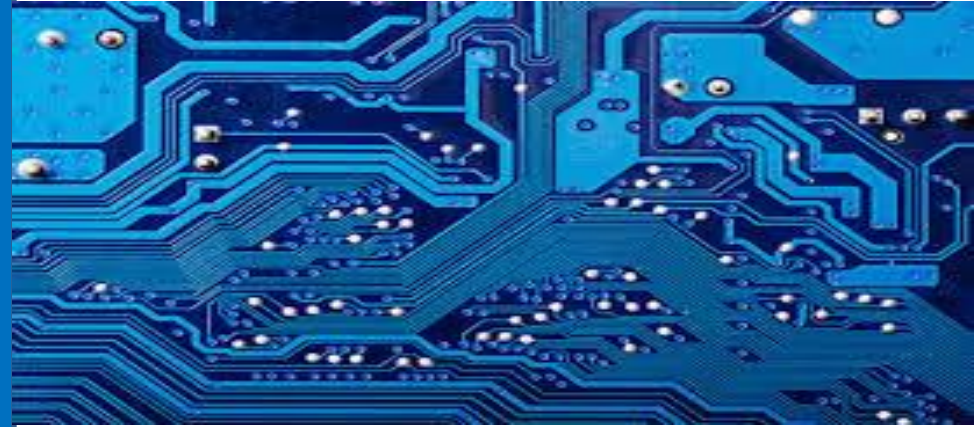


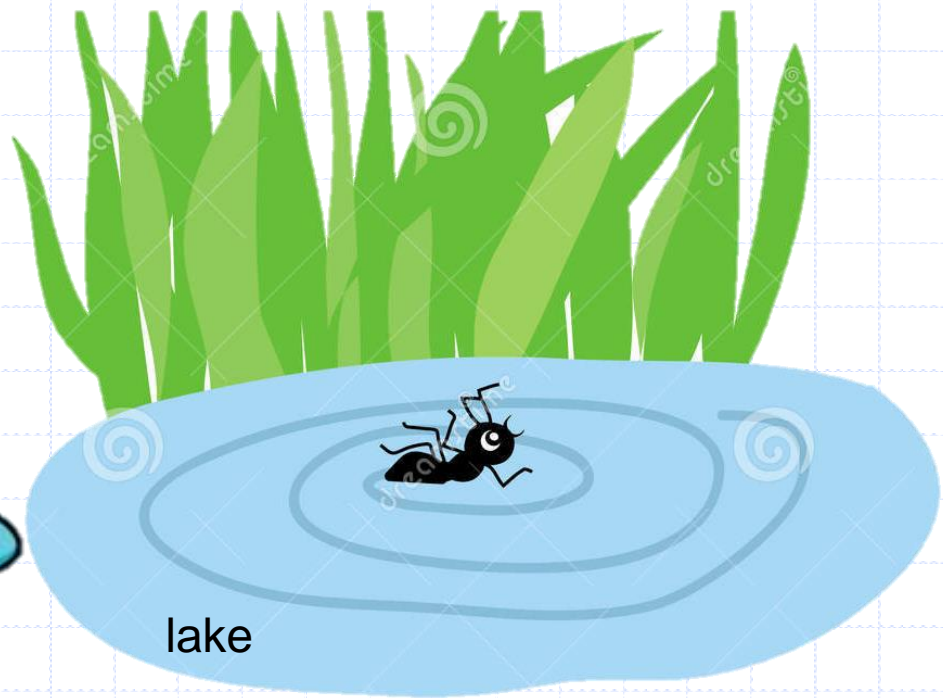
## Lecture 1

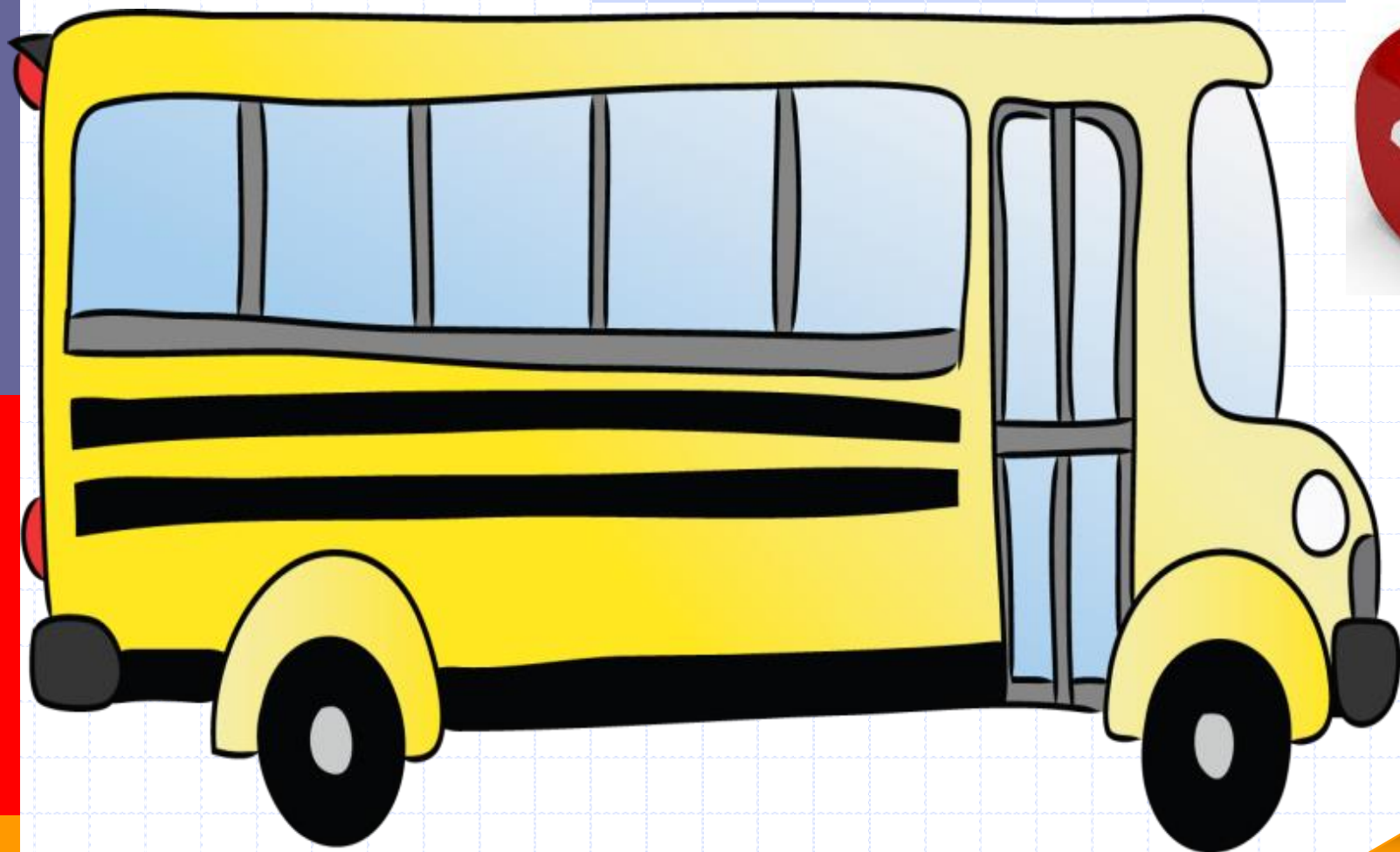
# Course Introduction

**Algorithms,  
Design and  
Analysis**



**Assoc. Prof. Anwer Sayed**





**Introduction to  
algorithms, Algorithms  
phases**

**Representing algorithms  
using different techniques,  
The efficiency of algorithms**

# **Course Contents**



**Time efficiency-  
Asymptotic Notation**

**Big-O-Notation, Examples**

**Recursion and its  
complexity, Searching  
algorithms**

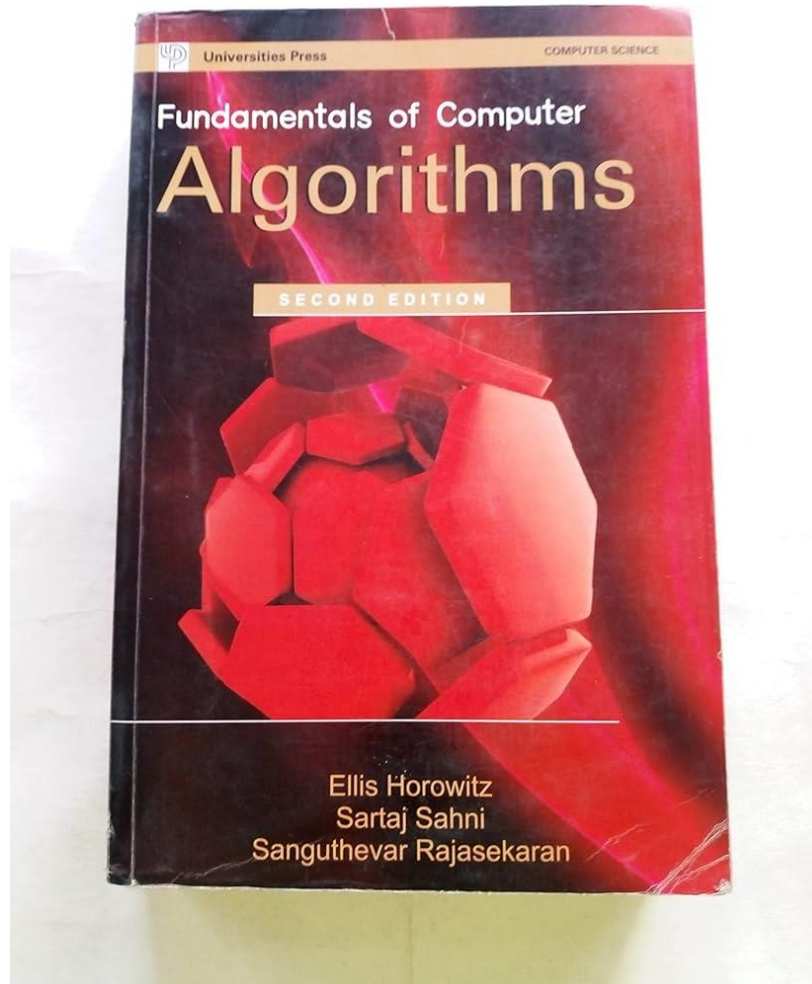
**Graph and tree  
algorithms, and other  
selected topics**

**Bubble sort, insert,  
selection sort algorithms**

**(\*) If time permits**

# Course Textbook

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharan, 3<sup>rd</sup> Edition University Press.





# Lecture Style and Recommendations

- **Lecture notes** may have **gaps** in them that need to be **filled in** while you are in lecture (i.e. good notes!)
  - Goal is to facilitate learning
  - Consider using blank back-side of slides for notes
- If you **miss** a lecture, you will need to ask others for their notes
  - You can ask me **follow up questions** once you have gone through those notes

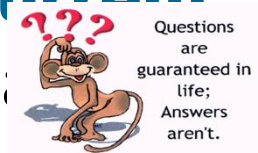


# Lecture Style and Recommendations

- **Get help!** – **Office hours** are a great time to learn. All I ask is that you be **knowledgeable** of your **ignorance!**



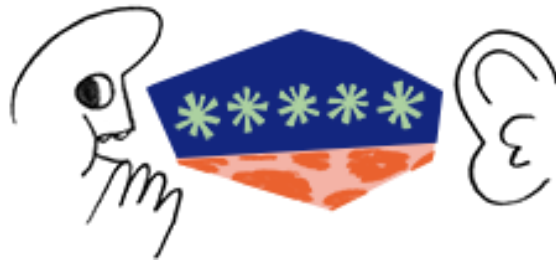
- **Be prepared for each lecture** – Attend each lecture having **read the notes** from the previous lecture, and having read the relevant **text** for the **current** lecture. Come to lecture prepared to think **deeply**!



- **Ask Questions!** – I will make **an effort** to periodically stop and see if everyone understands the lecture material. However, you should stop me at any time if you have any questions.
  - If you are **confused** about something, chances are so is someone else.

# Feedback

- Your comments and feedback are appreciated!



- Select my e-mail address under contacts, and send me an **anonymous** e-mail detailing your suggestion or concern.





# Exams and Cheating



**Exams are closed-book.  
HOWEVER**

**Formula sheet will be provided at midterm and final exams  
(if needed)**

**Old Exams and their solutions are on course website**

**Cheating results in 0 grade and academic dishonesty  
Cheating Policy: Just don't**



# Answer Clarity

## ★ You **must** present your answer clearly

- Answers with **units** are to be boxed and **right justified**.
- Show **supporting work** before the boxed answer with clearly shown steps of how you arrived at the answer.
- Grade reduction will occur for **sloppy** work.

## ★ Example of preferred presentation

Problem 1:

Drawing

Equation(s)

Answer = .....



# Course Assessment/Evaluation

- **Examination (40%)**
- **(\*) Coursework (30%)**
- **(\*) Midterm (30%)**



**(\*) subject to change**

# Lecture Outcomes

- Understand the definition and principles of algorithms and the steps of making an algorithm.

# **Let's enjoy discovering the secrets of Algorithms Course!**



**I'm hoping to generate insight and interest – not pages of  
equations!**

# Definition



# Historical Background

- ❑ The word Algorithm comes from the name of the Persian author, “ Abu Ja’afar Mohammed Ibn Mousa Al Khawarizmi (825 A.D)”.
- ❑ Al Khwarizmi wrote several books in Mathematics and Algebra.



## Linear and quadratic equations: Some of the first algorithms



## تعريف الخوارزمية Definition of an Algorithm

- ❑ The word Algorithm means “a process or set of rules to be followed in calculations or other problem-solving operations”. Therefore, Algorithm refers to a set of rules/instructions that step-by-step define how a work is to be executed in order to get the expected results
- ❑ An Algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output. Algorithms are generally created independent of underlying languages, i.e., an algorithm can be implemented in more than one programming language.
- ❑ An algorithm is a well-ordered collection of unambiguous and effectively computable operations that when executed produce a result and halts in a finite amount of time.



## Why We Study Algorithms?

- ❑ Easier to solve and code the problem

The programming process is a complicated one. You must first understand the program specifications, of course, and then you need to organize your thoughts and create the program. This is a difficult task when the program is not trivial (i.e., easy). You must break the main tasks that must be accomplished into smaller ones in order to be able to eventually write fully developed code.

Writing an algorithm will save you time later during the construction & testing phase of a program's development.

- ❑ To be familiar with different strategies and approaches for solving different problems.
- ❑ To Find out which one of the known algorithms solves the problem in an efficient manner.



# Characteristics of an Algorithm خصائص الخوارزمية

- Correctness
- Clear and Unambiguous – Algorithm should be clear and unambiguous. Each of its steps (or phases), and their inputs/outputs should be clear and must lead to only one meaning.
- Input – An algorithm should have 0 or more well-defined inputs.
- Output – An algorithm should have 1 or more well-defined outputs and should match the desired output.
- Algorithms are well-ordered.
- Finiteness – Algorithms must terminate after a finite number of steps.  
**It halts in a finite amount of time.**
- Feasibility – Should be feasible with the available resources.
- Algorithms have effectively computable operations.



# **Steps For Developing Algorithms**

# Prerequisite

- ❑ In order to write an algorithm, the following things are needed as a prerequisite:
- The **problem** that is to be solved by this algorithm.
  - The **constraints** of the problem that must be considered while solving the problem.
  - The **input** to be taken to solve the problem.
  - The **output** to be expected when the problem is solved.
  - The **solution** to this problem, in the given constraints.
  - Then the algorithm is written with the help of above parameters such that it solves the problem.



# Steps for Developing an Algorithm

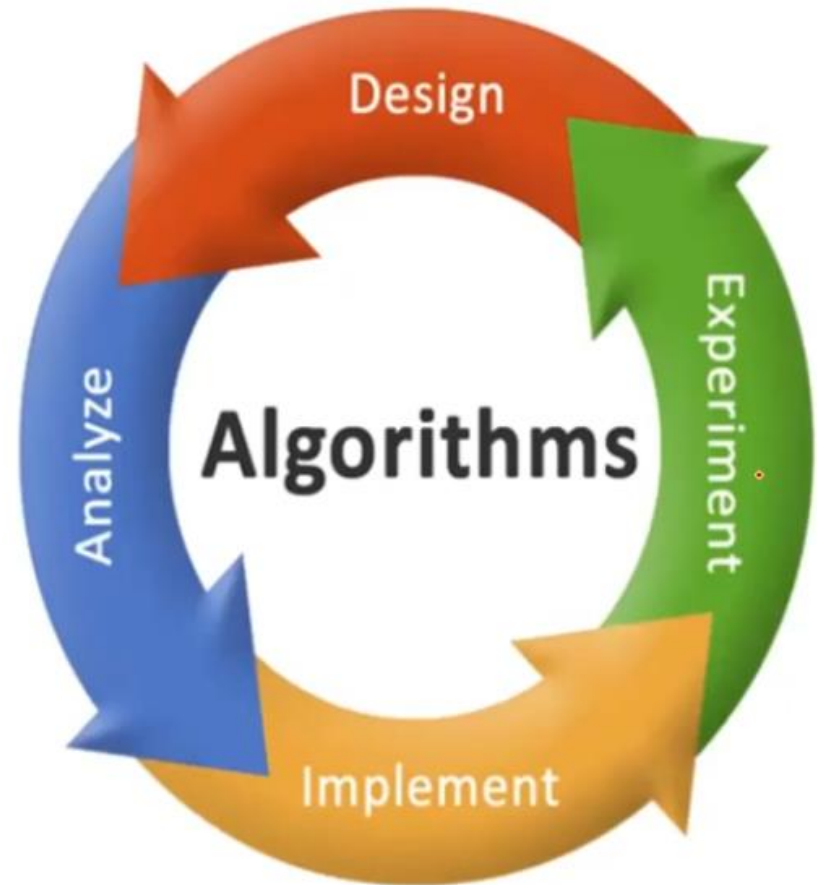
1. **Define the problem:** State the problem you are trying to solve in clear and concise terms.
2. **List the inputs** (information needed to solve the problem) and the outputs (what the algorithm will produce as a result)
3. **Describe the steps needed to convert or manipulate the inputs to produce the outputs.** Start at a high level first and keep refining the steps until they are effectively computable operations.
4. **Test the algorithm:** choose data sets and verify that your algorithm works!



# Algorithm Phases مراحل الخوارزمية

## ❑ Algorithm Phases

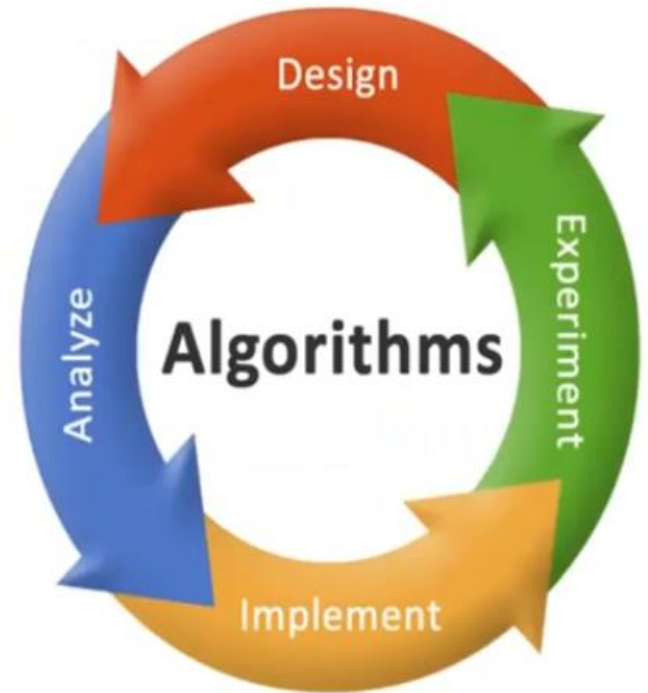
- ❖ Design
- ❖ Analyze
- ❖ Implement
- ❖ Experiment



## Algorithm Phases مراحل الخوارزمية

### □ Design Phase

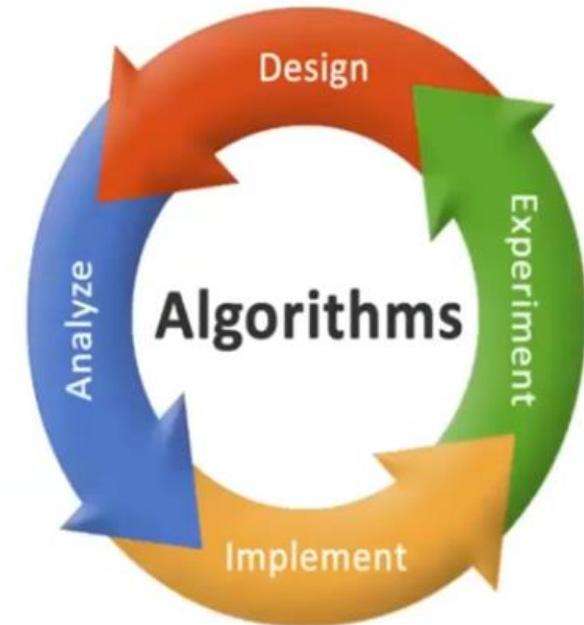
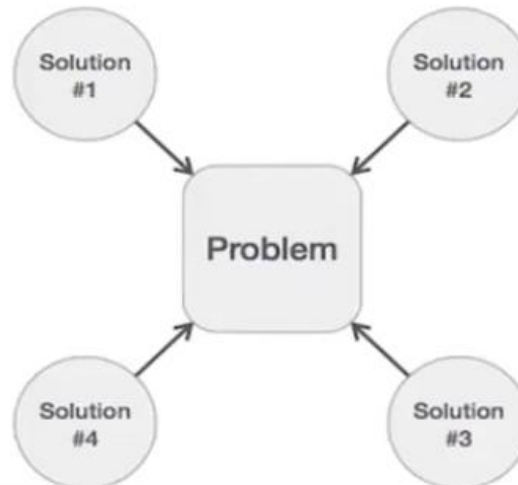
- The first stage is to identify the problem and fully understand it.
- Consultation with people interested in similar problems



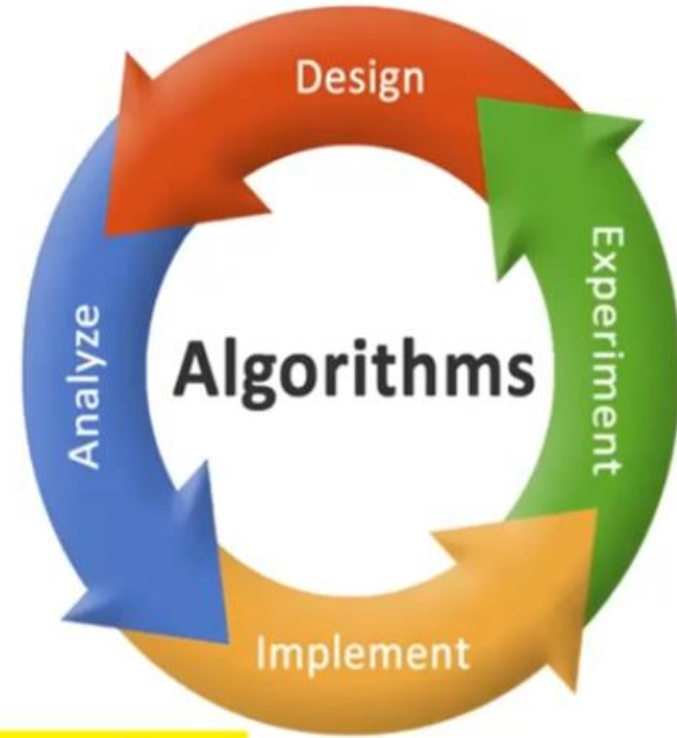
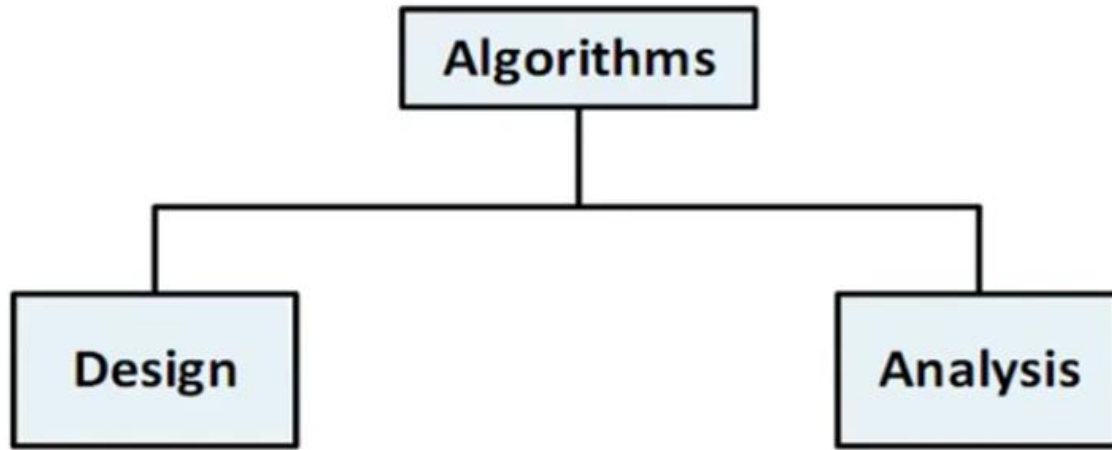
# Algorithm Phases

## Algorithm Analysis

- Analysis of an Algorithm is the theoretical study of computer program performance and resource usage
- Many solution algorithms can be derived for a given problem.
- The next step is to analyze those proposed solution algorithms and implement the best suitable solution.



# Algorithm Phases مراحل الخوارزمية

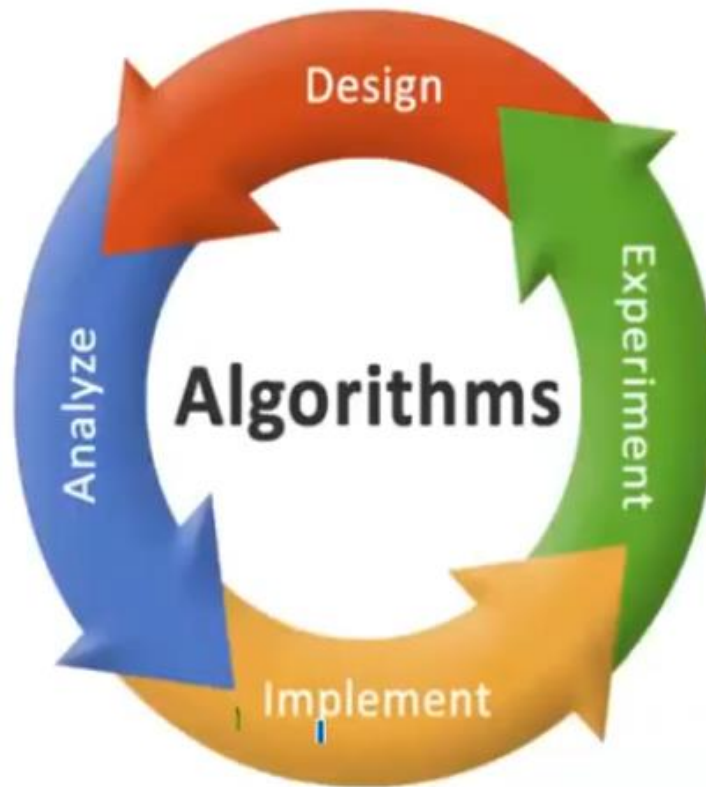


Methods and techniques which yield a good and a useful algorithms to solve the problem.

Mathematical Comparison of algorithms without actually implementing it.



# Algorithm Phases

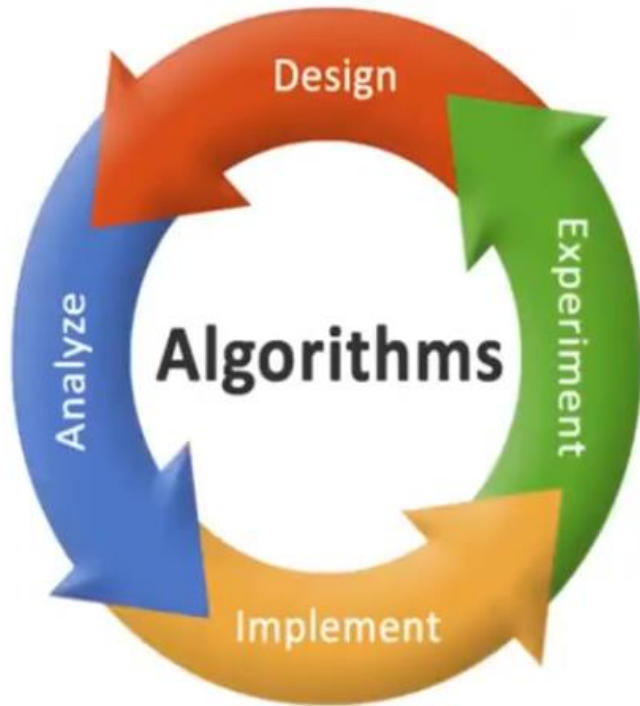


- **Implement**

➤ Writing and coding the algorithm

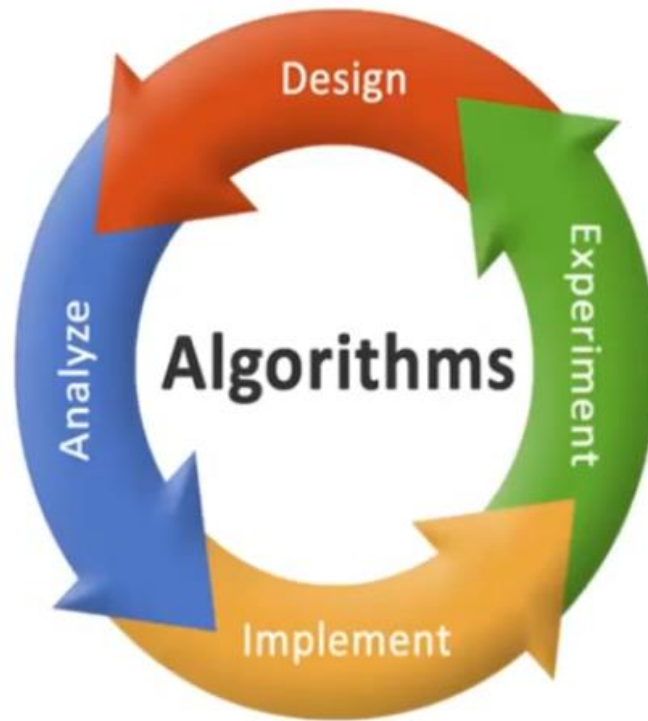


# Algorithm Phases



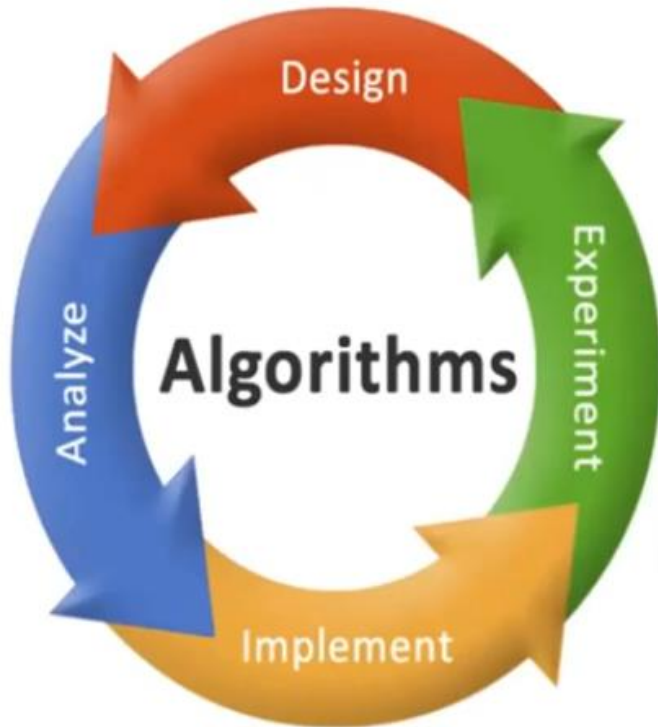
- Experiment
  - Experiment with different cases and variables

# Algorithm Phases مراحل الخوارزمية



- The design and analysis of algorithms is a circular process.

# Algorithm Phases



- The Quality of good Algorithm

- Time
- Memory
- Accuracy
- Sequence
- Generality

# Operations in an Algorithm

## ❑ Sequential operations.

A sequential operation carries out a single well-defined task. When that task is finished, the algorithm moves on to the next operation.

## ❑ Conditional operation.

A conditional operation is the “question-asking” instructions of an algorithm. It asks a question and then selects the next operation to be executed according to the question-answer

## ❑ Iterative operations.

An Iterative operation is a “looping” instruction of an algorithm. It tells us not to go on to the next instruction, but, instead, to go back and repeat the execution of a previous block of instructions



Any  
Questions ?

