# General Notes

## What is an algorithm?

An algorithm is a set of well-defined rules or a recipe to solve a problem. For example, if we have a bunch of tasks that need to be completed according to their deadline, an algorithm will solve that problem.

Routing and Communication depend on the shortest path algorithm.

Algorithms are used in quantum mechanics and finance.

## Pattern for Pseudocode

* Define a computation problem.
* Define Input
* Define output.
* Solution that transforms input to output

## The algorithm Designer's Mantra

"Perhaps the most important principle for the good algorithm designer is to refuse to be content" Aho , Hopcroft and Ullman (the Design and analysis of computer algorithm, 1974)

Can we do better ?

## How to Design an Algorithm

### The scientific method (Algorithms 4th edition, Sedgwick et al)

* Observe some features of the natural world, generally with precise measurements.
* Hypothesize a model that is consistent with the observations.
* Predict events using the hypothesis.
* Verify the predictions by making further observations.
* Validate by repeating until the hypothesis and observations agree.

Understand the problem.

Understand the needed data structure in terms of input, output, and middleput

Iterate

## What is the mathematical proof?

The proof is verifying that a proposition is true through a series of steps called "logical deductions" from a base set of axioms (set of self-evident truths), they are the starting point of logical reasoning. poor axioms (shaky truth) can lead to unreliable mathematical reasoning. (paraphrased from Mathematics for Computer Science, Eric Lehman, and Tom Leighton)

\*\*\* Propositions

In our logic system, Propositions are binary statements that are true or false. however, in another system, a likelihood can be attached to the proposition (completely true or completely false). A particular case of proposition is called a predicate whose truth depends on the value of one or more variables.

implications (==>) have the following truth tables

P Q P ==> Q

T T T

T F F

F T T (accepted mathematical convention)

T T T

for if and only ( <==>)

P Q P==>Q Q ==> P P<==>Q

T T T T T

T F F T F

F T T F F

F F F F T

\*\* Axioms

an axioms is a proposition that you believe is true . Axioms should be consistent and complete

* Consistent, no proposition is self-contradictory
* Complete, if every proposition can proved or disproved .

\*\* Logical deductions

Also called inference rules, combines axioms and true proposition to generate more true propositions

* **Modus Ponens ,**if P is true and P==> Q is true it means Q is true
  + **Tautology** is one true proposition and, each tautological proposition there is an associated inference rule

## Concepts

### loop invariant

a loop invariant is a property of an algorithm or functionality that satisfied the following conditions

Initialization: is true prior to the first iterations of the loop

Maintenance : is true before the loop iterations and before the next iterations

Termination: When the loop terminates, the invariant should help establish the correctness of the algorithm ( as an example, reduce (the inclusive sum of the array's element is a loop invariant)

## Guiding Principles of for the Analysis of Algorithms

In our quest in finding a fast algorithm “whose worst-case running time grows slowly with the input size”, according to Roughgarden, Linear time is the holly grail. Also, the Fast algorithm can be considered for free primitives, like merge sort, which are used in preprocessing the data.

## Principle 1: Worst Case Analysis

* Mathematically more Tractable than other kind of analysis
* Fewer assumptions
* Doesn’t require domain expertise compared to average case analysis.

## Principle 2: Ignoring Constants

* For mathematic tractability
* constants can vary from one platform to another.

## Principle 3 : Asymptotic Analysis

* Asymptotic analysis is studying the rate of growth of running time with input lengt