# Algorithms – Theory & Concepts

## GENERAL INTRODUCTION TO ALGORITHMS

## SEARCHING ALGORITHMS

* Searching Algorithms are designed to check for an element or retrieve an element from any data structure where it is stored.
* Based on the type of search operation, these algorithms are generally classified into two categories:
  + **Sequential Search**: In this, the list or array is traversed sequentially, and every element is checked. For example: Linear Search.



**Linear Search to find the element “20” in a given list of numbers**

* + **Interval Search**: These algorithms are specifically designed for searching in sorted data-structures. These type of searching algorithms are much more efficient than Linear Search as they repeatedly target the center of the search structure and divide the search space in half. For Example: Binary Search.



**Binary Search to find the element “23” in a given list of numbers**

### Linear Search

* **Linear search is searching technique that searches for any item serially from the first element until the item is found in any collection. It scans the element one-by-one until the item that is searched matches the item currently scanned.**
* **Complexity:**
  + **Worst case: O(n) => if the item to search is present at last index**
  + **Best case: O(1) => if the item to search is present at first index**

**Problem Statement :** Given an array arr[] of n elements, write a function to search a given element x in arr[].

### 

#### General Purpose & Motivation

This course is dedicated for CGI Trainees with an appetite to learn the world of Big Data & Cloud.

#### Objectives

* Learn the main concepts related to big data.
* Understand the purpose of using Hadoop and how it works.
* Understand all Hadoop-based technologies: Hive, Pig, …
* Understand how Spark Can be used to analyse big data sets.
* Discover the most used ETL within enterprises: Nifi, Talend

#### Prerequisites

* Windows System
* Linux Knowledge
* Fundamental concepts of programming: Functional Programming, Object Oriented Programming

#### Duration of training

#### Philosophy

* **40%** Theory
* **10%** Quizzes
* **50%** Hands-on (Coding Exercises & Practice Tests)

## INTRODUCTION TO BIG DATA

## APACHE HADOOP

### Introduction to Hadoop

#### What is Hadoop?

#### Why Hadoop?

#### History of Hadoop

#### Core Hadoop

#### Hadoop Ecosystem

#### Hadoop features

### Hadoop File Distributed System (HDFS)

##### Introduction to HDFS

##### HDFS Architecture

##### HDFS Commands

### MapReduce

##### Introduction to MapReduce

##### MapReduce Architecture

### YARN

##### Introduction to YARN

##### YARN Architecture

### Hadoop Installation & Configuration

##### Hadoop Distributions

##### Hadoop Standalone Installation

### Hadoop Limitations

##### Hadoop Limitations

## APACHE HIVE

## APACHE PIG

## APACHE OOZIE

## APACHE SQOOP

### Introduction to Sqoop

#### What is Sqoop?

#### Why Sqoop?

#### History of Sqoop

#### Sqoop features

### Sqoop Architecture

### Sqoop Installation & Configuration

### Sqoop Demo

## APACHE R CONNECTORS

## APACHE HBASE

## APACHE ZOOKEEPER

## APACHE FLUME

## APACHE SPARK

### Introduction to Spark

#### What is Spark?

#### Why Spark?

#### History of Spark

#### Core Spark

#### Spark Ecosystem

#### Spark features

### Spark Architecture

##### Spark RDDs

##### Spark DataFrames

##### Spark DataSets

### Spark Installation & Configuration

### Spark Demo

## APACHE KAFKA

## NIFI

## TALEND

## TABLEAU & KIBANNA FOR DATA VISUALIZATION

## INTRODUCTION TO CLOUD

## USE CASES & PROJECTS