



# Alt School Africa Class Notes[Key Concepts of Computer Science]

## Introduction

Trust me, this chapter is a long one but not too difficult to understand; A lot of concepts there were explained with relatable real-life scenarios but I will also use words we could relate to also, Let's go.

- **Algorithms & Data Structures**

- **Big 'O' Notation:** basically deals with **TIME** and **Worst-case Scenario**, It's just like trying to know what happens(time it takes) if you "overdo" something. Take note:
  - i. **IF YOUR INPUT DOES NOT AFFECT YOUR OUTPUT TIME(delivery time) THAT IS  $O(1)$ [constant time complexity].**
  - ii. **IF YOUR INPUT AFFECTS YOUR OUTPUT TIME EITHER POSITIVELY or NEGATIVELY THAT IS  $O(n)$ [Linear time complexity]**

**Why do we need it?** To be able to know if our algorithm is scalable and efficient.

- **Sorting Algorithm:** as the name implies, **Sorting Algorithms** are methods of reorganizing a large number of items into some specific order such as highest to lowest, or vice-versa, or even in some alphabetical order.

Types of sorting algorithm:

- Bubble Sort
- Merge Sort
- Insertion Sort
- Selection Sort
- Quick Sort

**YOU DON'T NEED TO DEEPLY UNDERSTAND THEM NOW[for the scope of this course] BUT YOU CAN LOOK THEM OUT LATER.**

- **Recursion:** occurs when a thing is defined in terms of itself or of its type, Like repetition. For example, "Quoting a tweet with the same tweet".



- **BIG DATA:** Just like it implies, BIG DATA if your product deals with a lot of data you definitely need a specialized means of working with them.
- **Data Structures:** They are a specialized means of organizing and storing data in computers in such a way that we can perform operations on the stored data more efficiently. Types of data structures we have:
  - Array
  - Tree
  - Stack
  - Queue
  - Graph
  - Hash Table

- vii. Linked List
- viii. Heap

NOTE: for the scope of this course you don't need an in-depth knowledge of them, you can look into them later though.

- **Artificial Intelligence**

- **Greedy Algorithm:** as the name implies, it's greedy therefore it doesn't think long-term, it chooses **what's best at the moment**.
- **Hill Climbing:** A hill-climbing algorithm is a local search algorithm that moves continuously upward (increasing) until the best solution is attained. This algorithm comes to an end when the peak is reached. i.e It keeps trying new ways to get the best solution.
- **Simulated Annealing:** In simulated annealing the procedure is the same as the hill-climbing but you have a probability of accepting a solution that is worse than the previous one. In simulated annealing, you keep trying different steps before moving to the next one so as to get the best solution.
- **Dynamic Programming:** It's just a fancy name for saying that when you are breaking the problem down into subproblems, you'll store their values. Next time, if you encounter the same subproblem, you'll just reuse the stored value instead of recalculating.
- **Machine Learning:** Machine Learning as the name suggests, involves making the machine learn.
  - a. **Learn what?** Learn the decisions to be taken on a particular case, just like humans do. Hence, Machine Learning is a branch of Artificial Intelligence.

b. **Learn How? How do humans learn something?** By practice, i.e by applying your knowledge many times in different cases.

- **P vs NP problems:** NP Complete problems are problems where you can easily computationally determine the correctness of a solution but not its solution.

- **Computer Architecture and Engineering**

- **How computers work?** Computers work by adding complexity on top of complexity. When you use your Laptop, you don't necessarily have to understand how the Computer works. The complex details are hidden.
- **Halting Problem:** That the only way to know the answer to a calculation is to actually do the calculation. You cannot know the answer without going through the process of doing the calculation. **The Halting problem is just a particular example of this, where the "calculation" is to decide if the calculation process itself will stop or halt.**

- **Concurrency**

**Concurrency is a fancy way of saying simultaneous occurrence of events.**

- **Parallelism:** Parallelism allows 2 or more tasks to run at the **same time**, provided that the machine has multiprocessing capability.
- **Race Condition:** is a situation where several processes access and manipulate the same data concurrently and the **outcome of the execution depends on the particular order in which the access takes place.**
- **Mutual Exclusion:** simply explains **if two things cannot happen at the same time**, they are mutually exclusive. If you turn right, you cannot turn left: turning left or right are mutually exclusive.
- **Semaphore:** A semaphore is a programming concept that is frequently used to solve multi-threading problems.

Think of semaphores as bouncers at a nightclub. There are a dedicated number of people that are allowed in the club at once. If the club is full no one is allowed to enter, but as soon as one person leaves another person might enter. It's simply a way to limit the number of consumers for a specific resource.

- I. **Binary Semaphore:** A "binary semaphore" can have two values only; one to represent that a process/thread has a higher priority and other ones should wait, the other indicating the critical section is free.
  - II. **Counting Semaphore:** counting semaphores take more than two values, they can have any value you want. The max value N they take allows N process/threads to access the shared resource simultaneously.
- **Deadlock:** What it means literally, everything is coming to a halt because it cannot process. **Deadlock** is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.



The image above is a good representation of what Deadlock means.

Note: Concurrency is the ability of your program to **deal (not doing)** with many things at once.

- **Computer Security**

- **Computer Hacking:** I know this phenomenon is not strange, it's basically gaining access to another person's private information through different means. Types of hacking:
  - **Brute-force attack:** It's basically guessing what the password could be by trying out common hints like your name, date of birth e.t.c.

- **Social Engineering:** It's more of psychology, This process is tricking users into revealing their private information.
- **Security Exploit:** This is basically just looking for security loopholes in the system to take advantage of.
- **Trojan Horse:** A trojan horse is a malware program that **pretends** to be helpful but runs malicious code in the **background**.
- **Rootkit:** Rootkit gains access into your computer by pretending to fix a problem but they create a problem themselves to fix but steals access to your computer.
- **Distributed-Denial-of-Service Attack(DDoS):** DDoS Attack tries to bring a site or service down by flooding it with visitors until they can not handle the service.
- **Cryptography:** Cryptography is the study and application of secure communication(Encryption). The commonly used cryptographic protocols are:
  - **Symmetric cryptography:** This type of encryption that uses the same key to encrypt(lock) and decrypt(unlock) data.
  - **Asymmetric cryptography:** This type of encryption uses two separate keys for encryption and decryption — a public key(address) and a private key(password).

## ● **Software Development Methodologies**

This is like a workflow i.e how Softwares are built for easy documentation and laid out plans. Some ways are listed below:

- **Waterfall Development:** as the name implies, is a sequential development process that flows like a waterfall through all phases of a project (analysis, design, development, and testing, for example), with **each phase completely wrapping up before the next phase begins**.
- **Agile Development:** Agile Methodology meaning a practice that promotes **continuous iteration of development** and testing throughout the software development life cycle of the project.

END.

---

Please feel free to connect with me on twitter **@A\_A\_Adedotun**, if you have anything that could help us do better in our exams, let's study together

Success in our exams.