# Course Structure: Introduction to Data Structures and Algorithms

Purushottam

November 8, 2023

## Course Overview

This course is designed to provide solid foundation to Data structure and algorithm. To provide a goof command over STL. To provide subject knowledge required to solve questions in internship and placement coding rounds.

## Course Duration

- Course Type: Semester-long
- Duration: 8 weeks

## Course Objectives

1. Understand Fundamental Concepts.
2. Implement Data Structures.
3. Analyze Algorithms.
4. Problem Solving.
5. Real-World Applications.

## Course Outline

### Day 1: Vectors

- Declaring and Initializing vector
- Initializing by size and dafault value
- Index based looping , and range based looping
- Iterators and iterator based looping

- push_back , insert and erase method
- vectors as arugument to functinos and as return type
- pass by value and pass by reference

### Day 2: Fundamentals of data structures

- Pointers
- Structures
- Classes

### Day 3: Time and Space Complexity

- Meaning of time complexity
- loop time complexity, recursion time complexity
- linear time complexity
- exponential time complexity
- logarithmic time complexity
- loglinear time complexity
- Space complexity

### Day 4: Linked List

- Node and Linked List meaning
- Insert Function implementation
- Print function implementation
- Erase function implementation
- doubly linked list implementation
- circular linked list implementation
- use and time complexity analysis

### Day 5: std::algorithm

- Linear Search algorithm
- Binary search algorithm
- in-built searching for vector
- Bubble Sort algorithm
- Selection Sort algorithm
- Insertion Sort algorithm

### Day 6 : std::algorithm

- Merge Sort algorithm
- Quick Sort algorithm
- Count Sort algorithm
- In-built sorting
- pair
- comparator based sorting

### Day 7: Sets

- Meaning of sets
- std::set and its methods
- std::multiset and its methods

### Day 8: Wrappers

- Stack - implementation
- std::stack
- Queue - implementation
- std::queue

### Day 9: Binary Trees

- Implementation
- dfs
- bfs

### Day 10: Binary Trees

- Height of a node
- Diameter
- Inorder traversal
- Preorder Traversal
- Postorder Traversal

### Day 11: AVL Trees

- Balance of a node
- Balanced Binary Trees
- Rotations
- AVL Trees

### Day 12: Red Black Trees

- Red Black Trees - concept
- Red Black Trees - implementation

### Day 13: Dynamic Programming

- Recursion vs loops
- Memoization
- Dynamic Programming - top to bottom and bottom to top
- Space reducing DP

### Day 14: Graphs

- Graph - Meaning, related terms
- Adjacency Matrix representation
- Adjacency List Representation

### Day 15: Graphs

- Depth-First-Search
- Breadth-First-Search
- Number of component

### Day 16: Graph

- Cycle Detetion in graphs
- Diameter of graph

### Day 17: Graph

- Trees
- Minimum Spanning Trees
- Prim's algorithm
- Kruskal's algorithm

### Day 18: Graph

- Minimum Distance
- Djkastra's algorithm
- Bellman-Ford Algorithm
- Negative weight cycle detection

### Day 19: Graph

- Topological Sorting concept
- Sort using dfs

### Day 20: Hashing and Hash Table

- Hash Function
- Hash Table
- Hashing

### Day 21: String Algorithms

- KMP algorithm
- robin algorithm

### Day 22: 2D- DP

- Coin Problem
-

### Day 23: Skip List

- skip-list implementation

### Day 24: Heaps

- Heap - implementation
- HeapSort
- Priority queue
- minheap and maxheap

## Teaching Methods

- Pre-lecture notes to give intution
- Lectures to introduce concepts
- Coding examples and demonstrations
- Homework practice problems
- Assignments
- Placement round question banks

## Textbooks and Resources

- A competitive Programmer's Handbook
- Class Notes

## Prerequisites

- Basic programming knowledge and understanding of fundamental mathematics.

## Final Thoughts

Data structures and algorithms are a crucial part of any computer science curriculum. Ensure that the course offers a balance between theory and practical implementation. Students should leave the course with a strong foundation in these core concepts, ready to tackle more advanced topics in the future.