Quick Revision

A flashback over what has been learnt with highlights on some important topics covered

Menu

- Data structures
- Motivation of studying data structures
- Language to study data structures
- **Abstraction**
- **Huffman coding & priority queues**
- **Information hiding**
- **Encapsulation**
- **Efficiency in space & time**
- **Static vs dynamic data structures**

Menu

- Overview of Data Structure Programming topics
- Programming with Libraries
- **Collections**
- **Programming with Lists of objects**

Menu

- Collections and List
- **Using List and ArrayList**
- Iterators

Menu

- More Collections
- **Bags and Sets**
- **Stacks and applications**
- **Maps and applications**

Menu

- **Queues and Priority Queues**
- **Classes/Interfaces that accompany collections**
  - **Iterator**
  - **Iterable**

Menu

- **Iterators and Iterables**
- Sorting collections
- **Comparators and Comparables**

Menu

- Comparators
- Exceptions
- **Implementing Collections:**
  - **Interfaces, Classes**

Menu

- Implementing Collections:
  - Interfaces, **Abstract Classes**, Classes

Menu

- **Implementing ArrayList**:
  - Iterators
  - **Cost of adding and removing**

Menu

- Cost of operations and measuring efficiency
- **ArrayList: remove, add**
- **ArraySet: contains, remove, add**

Menu

- **recursive functions**
- factorial function
- fibonacci function
- recursion vs iteration

Menu

- Testing collection implementations
- **Queues**
- Motivation for linked lists
- **Linked structures for implementing Collections**

Menu

- **Linked structures for implementing Collections**
- A collection class
- Adding/Removing from the front

Menu

- **A Stack using a Linked List with a header**
- **A Queue using a Linked List with a header**

Menu

- **Cost of ArraySet operations**
- **Binary Search**
- **Cost of SortedArraySet with Binary Search**

Menu

- Binary Search
- Sorting
  - approaches
  - selection sort
  - insertion sort
  - bubble sort
  - analysis
  - fast sorts

Menu

- Sorting
  - **Design by Divide and Conquer**
  - **Merge Sort**
  - **QuickSort**

Menu

- Analysing Fast Sorting Algorithms
  - MergeSort
  - QuickSort

Menu

- Introduction to Trees
  - What are trees?
- **Binary Tree**
- **General Tree**
- Terminology
- Different Types of Tree
- **Tree Ordering**
- **Trees and Recursion**
- What are they useful for?
  - Tic Tac Toe example
  - Chess
  - Taxonomy Tree
  - Decision Tree

Menu

- **Maps**
- Search lists
- **Binary search trees**
- **Tree traversal**
  - **Preorder**
  - **Inorder**
  - **Postorder**
- **Balanced Search Trees**
  - **AVL Trees**

Menu

- **Implementing Binary Trees**
- **Implementing General Trees**

Menu

- **Hash tables**
- **Comparison among various search mechanisms**
- Table size
- **Hash function**
- Modular hash function
- Hash function examples
- **Collisions**
- Applications

Menu

- Basic definitions of graph theory
- **Properties of graphs**
- **Paths**
- **Trees**
- Digraphs and their applications, network flows

Menu

- Paths
- Connected graphs
- **Incidence matrix and adjacency matrix of a graph**

Menu

- Trees and forests
- Spanning trees
- **Minimum spanning tree**
- **Greedy algorithm for determining a minimum spanning tree**
- **Shortest path problem**