CS2233 Data Structures

This is a standard undergraduate course in Data Structures. A natural continuation to Introduction Data Structures from last semester, in this course we will see more nontrivial data structures that can be used in computational problems.

Administrivia

Instructors

- Subrahmanyam Kalyanasundaram subruk@cse.iith.ac.in
- Karteek Sreenivasaiah karteek@cse.iith.ac.in

Grading scheme:

- Mainly programming assignments: around 60% (6 or 7 assignments)
- Short Quizzes: Around 6 or 7 such short quizzes (3-4 simple questions each) 20%
- Descriptive Exams: Around 2 or 3 such exams 20%

Lecture plan:

- First lecture on Sep 3. It will be a meet and administrivia explanation.
- Lectures will be of two types.
    - For some topics (most initial topics), we shall point you to available video lectures, with online meetings for Q&A.
    - For the other topics, we shall make available pre-recorded video lectures, with online meetings for Q&A.

- We will use Google Classroom to communicate with the class. However, the programming assignments will be conducted through another platform (to-be announced).

Reference texts:

- CLRS Book: "Introduction to Algorithms" by Cormen, Leiserson, Rivest and Stein.
- Lecture Videos by Prof. Naveen Garg: Link
- Lecture Videos by Profs. Erik Demaine and Srini Devadas: Link

High Level Set of Topics to be Covered

- Motivation. Why do we need sophisticated data structures?
- Binary Search Tree: Search, Insertion, Deletion, Successor, Predecessor
- Red-Black Trees: Properties, Insertion, Deletion (or/and AVL trees)
- Connection between Randomized Quicksort and Insertion in BST: Expected average node depth of a ranomly built BST, Expected height of a randomly built BST
- (2,3,4)-Trees and connection to red-black trees, B-Trees.
- Heaps: Binary max and min heaps, Heapify, Build Heap.
- Graphs: Adjacency Matrix, Adjacency List
- Breadth First Search: Proof that BFS finds the shortest path, Time Complexity
- Dijkstra's Algorithm: Shortest Path in Weighted Graphs, Proof of Correctness
- Minimum Spanning Tree and Kruskal's Algorithm, Proof of Correctness, Time Complexity
- Disjoint Set Data Structure:
    - List Implementation: Union by Rank Heuristic
    - Implementation using Disjoint Forests: Union by Size Heuristic, Union by Rank Heuristic, Path Compression Heuristic

- Amortized Analysis: Aggregate Method, Accounting Method, Potential Method
- Other topics (time permitting, also based on popular choice)
    - Skip Lists
    - Hashing
    - Binomial/Fibonacci Heaps