ADSA: Advanced Data Structures and Algorithms

Semester: Jan – May 2020; Course Type: PE; Mode: Self-Study for Y16 SLI students of CSE and CCE branches;

Google Classroom Course Name: ADSA Jan 2020; Code: xgdtjh3

CIF: Shared

Text Book:

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. Introduction to algorithms, 3rd Edition, The MIT Press, 2009.

Evaluation Criteria: as mentioned in the CIF: (i) Mid-term 40% and (ii) End-Term 60%

Topics before Mid-Term:

[A]. For the month of Jan 2020:

1. Algorithm complexity and asymptotic notations;
2. Revision of BT, BST, AVL Tree Operations and their Time Complexities in BIG O notation;
3. Red-Black Trees, 2-3 Tree, B-Tree, Skip List, Heaps: Binomial and Fibonacci;
4. Data structures for disjoint sets: Union Find with applications;

[B]. For the month of Feb 2020 until Mid-Term:

5. Hashing: Fundamentals, Simple Uniform, Double Hashing, Universal & Perfect Hashing, Application;
6. Text Processing: Pattern Matching – KMP algorithm, Boyer Moore algorithm;
7. Tries- Standard Tries, Compressed Tries, Suffix Tries;
8. Implementation and Application of Text Processing Algorithms;

Units – 3 and 4 will be for study post Mid-term.

Home Assignment # 1: For the week of Jan 20th: No need to submit but practice;

1. What are the worst-case time complexity to (i) find, (ii) insert and (iii) delete a key in:
    a. Unsorted array, sorted array, BT, BST and AVL Tree: justify your answer;

Home Assignment # 2: For the week of Jan 20th: No need to submit but practice;

1. Perform the following "add a key-value" operations into an AVL-Tree. Whenever an imbalance has occurred and there is a need to perform a rotation, indicate the node that got into imbalance (i.e. BF is either +2 or -2), what type of rotation is required (SLL,SRR, DLR or DRL), and show the resulting AVL-Tree;
   Key-values: 150, 100, 50, 200, 175, 75, 60

2. What is worst-case time complexity of SLL, SRR, DLR, DRL (only rotation but NOT propagation);

Home Assignment # 3a: For the week of Jan 20th: No need to submit but practice;

1. Define a Red-Black-Tree (RBT), in terms of its characteristics.
2. The operation, "Insert a key-value into an RTB" is usually discussed by dividing this operation into different cases, based on the "current state of RBT". Precisely and concisely, discuss these different cases, with an example for each.
3. Perform the following "insert key-value" operations into an empty RBT. For each insert operation, state the case into which it falls and indicate the steps taken. Show the resulting RBT at the end of each insert operation.
   150, 100, 50, 175, 25, 75, 60, 160, 85, 70
4. What is the upper bound on the height of an arbitrary RBT in terms of the number of nodes "n".
5. What is an advantage of RBT over AVL-Tree, while performing insert and delete of a node (key-value)?

Home Assignment # 3b: For the week of Jan 20th: No need to submit but practice;

1. Define a 2-3-Tree (2-3-T), in terms of its characteristics.
2. The operation, "Insert a key-value into an 2-3-T" is usually discussed by dividing this operation into different cases, based on the "current state of 2-3-T". Precisely and concisely, discuss these different cases, with an example for each.

3. Perform the following “insert a key-value” operations into an empty 2-3-T. For each insert operation, state the case into which it falls and indicate the steps taken. Show the resulting 2-3-T at the end of each insert operation.
   50, 60, 70, 40, 30, 20, 10, 80, 90, 100