CS352 – Data Structures and Algorithms

Time: 3 Hours | Final Exam – Jan 21, 2019 | Total Marks: 75

Note:

- *No code is required for questions 1, 2 & 3*
- *Code is only required for questions 4-7. You are expected to write ADT functions, not the client code*
- *Use of mobile phones and calculators is not allowed*

Q1: Briefly answer following questions: {15}

a. Let's assume that we implement a Priority Queue using the Unsorted Linked List ADT. What will be its disadvantages in comparison to Heap based implementation?

b. In which situation a binary search tree will not be efficient for search? Why?

c. Explain the multiplication method used in hashing. What are its advantages and disadvantages over division based method?

d. Compare the advantages and disadvantages of Adjacency Matrix and Adjacency List based implementations of Graph ADT.

e. If the data is almost sorted (only a few values are unsorted), should we use insertion sort or selection sort? Discuss why?

Q2: {6+2=8}

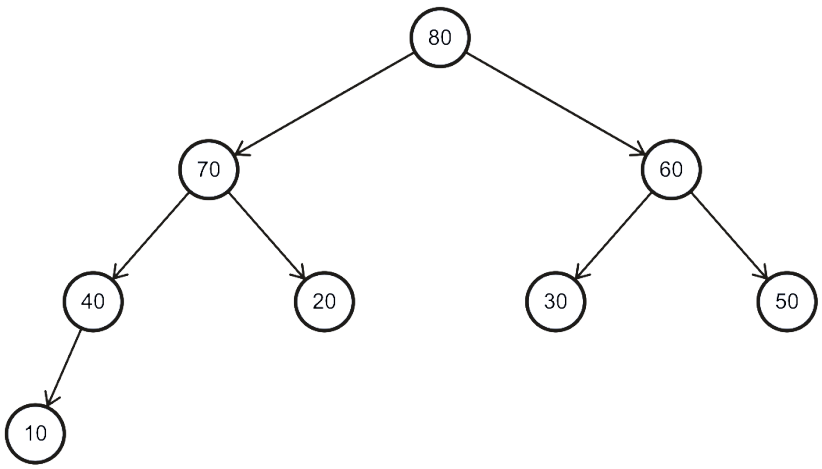
a. Convert the following infix expression into postfix: 2 - 7 / (3+4) * 6/2 + (4-3)

b. Evaluate the following postfix expression: 2 8 4 + + 5 + 6 2 * +

Q3: {12}

a. How would an empty heap structure look like after inserting the following values in the given order: 1, 5, 10, 15, 12, 11, 20

b. How would the following heap structure look like after deleting the following values in the given order: 80, 40, 70



Q4: Implement a function *DeleteGreater* in the UnsortedLinkedList ADT which takes a number as a parameter and deletes all the values greater than that number from the list. The list should be traversed only once. {10}

Q5: Implement *Insert* function for the Hash Table ADT which uses double hashing. {10}

Q6: Implement a function *CountLeaves* in the Binary Search Tree ADT which counts the leaf nodes in a binary search tree. {10}

Q7: Implement a function *DFS* in the Graph ADT which displays all the vertices in the graph traversed using Depth First Search. {10}