Indian Institute of Technology, Madras
Department of Applied Mechanics & Biomedical Engineering
Computational Tools: Algorithms, Data Structures and Programs - ID6105

Assignment set - II

To be submitted by: Monday 15^{th} September, 2025

1 Problem 1

Write a program to reverse the direction of a given singly-linked list. Thus, after the reversal all pointers should now point backwards. Your algorithm should take linear time ($O(n)$).

---

2 Problem 2

Solve the tower of Hanoi problem using three different stacks, each representing one towers. Solve recursively in moving n disks from tower A to tower C (using tower B). What is the time complexity of this recursion algorithm?

---

3 Problem 3

The problem with inserting elements into the binary tree structure is that it can lead to a skinny or unbalanced tree, which leads to linear scaling of operations instead of the logarithm scaling. In what follows we will try to mitigate this problem by considering a few operations which are part of self-balancing binary search tree.

(a) The idea is that, to maintain the balancing nature of the tree, one way to do this is to do rotation operations. One can do left or right rotations where the nodes get interchanged. Read about them and explain with examples.

(b) When does one use left rotation or right rotation?

(c) Given the following list of elements $[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$, draw the binary tree if one inserts element from the left end. Then use rotation operations to balance the tree and explain this process.

---

4 Problem 4

Entropy of binary trees: Balanced trees are useful so that the depth of the tree scales as a logarithm of number of nodes. We next want to quantify the balanced nature of a binary tree by measuring a quantity that we will call as Entropy (closer to information entropy). Given a binary tree T with n nodes, where each node v_i has a positive wieght $w_i(v_i)$, we define the local entropy as,

$$H(v_i) = -\sum_{c\in\text{children}(v_i)} \left(\frac{W_c}{W_v} \times \log_2\left(\frac{W_c}{W_v}\right)\right) \tag{1}$$

where W_c is the weight of the particular child c of v_i, W_v is the total weight of the subtree rooted at v_i (including v_i itself). Note that the above sum is over all the children of v_i. For leaf notes, since they do not have any children, $H(v) = 0$.

The total entropy of the binary tree then can be defined as,

$$S = \sum_{v_i\in T} H(v_i). \tag{2}$$

Now let us take an example with 15 random numbers lying between $[10, 99]$, avoiding repetitions, as the fifteen node binary tree that is to be constructed. The weights are the numbers themselves. Answer the following questions,

(a) Write a python code to construct a binary tree, and compute the entropy of individual nodes and the complete tree.

(b) Playing with the entrees, what does the entropy give a measure of?

(c) By shuffling the numbers around (or by a more thoughful algorithm), find the trees with maximum and minimum entropy.
