

Data Structures and Algorithms

Lab 03

Instructions:

- Work on this lab individually. Discussion is not allowed.
- Evaluation of tasks will be conducted in lab.
- Anyone caught being indulged in the act of plagiarism would be awarded an "F" grade in this lab.

Task 1: (35 Marks)

Create a singly linked list with only one variable to hold data in node. And implement the following functions **recursively**:

- Function print the data of list in reserve order
e.g. if this is the list 1->2->7->3->8, output must me 8, 3, 7, 2, 1
- Function to delete ith node.
- Append function to add values to the list.
- Delete by value function
- Function to empty the list by deleting all the nodes.
- Function to print alternate values of the list.
e.g. list 1->2->3->4->5->6->7->8 Output 1, 3, 5, 7
- Function for alternating split of Linked List

e.g. Input: 1 2 3 4 5 6 7

Output: 1 3 5 7

2 4 6

Input: 1 4 5 6

Output: 1 5

4 6

Task 2: (35 Marks)

Towers of Hanoi

Implement following **Recursive** function to display moves for n disk initially at source to target

void TOHmoves (int n, char src, char trg, char aux)

The Tower of Hanoi is a mathematical game or puzzle. It consists of three rods, and several disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

Instructor: Umm-e-Ammarah

Data Structures and Algorithms

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
3. No disk may be placed on top of a smaller disk.

Example:

Input: 2

Output: Disk 1 moved from A to B

Disk 2 moved from A to C

Disk 1 moved from B to C

Input: 3

Output: Disk 1 moved from A to C

Disk 2 moved from A to B

Disk 1 moved from C to B

Disk 3 moved from A to C

Disk 1 moved from B to A

Disk 2 moved from B to C

Disk 1 moved from A to C

APPROCH:

Take an example for 2 disks :

Let rod 1 = 'A', rod 2 = 'B', rod 3 = 'C'.

Step 1 : Shift first disk from 'A' to 'B'.

Step 2 : Shift second disk from 'A' to 'C'.

Step 3 : Shift first disk from 'B' to 'C'.

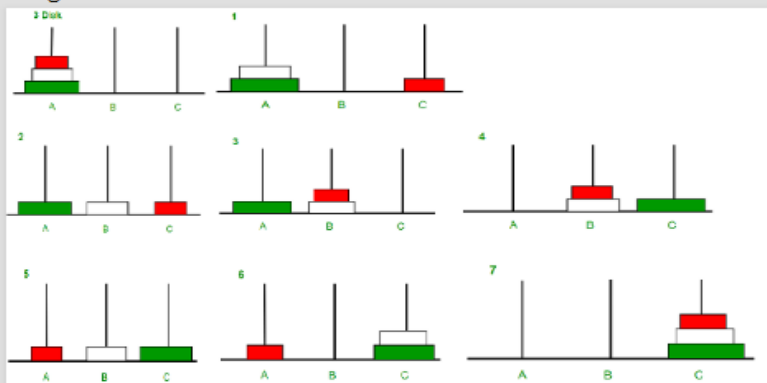
The pattern here is :

Shift 'n-1' disks from 'A' to 'B'.

Shift last disk from 'A' to 'C'.

Shift 'n-1' disks from 'B' to 'C'.

Image illustration for 3 disks :



Instructor: Umm-e-Ammarah