# Data Structures

## Data Structures and Algorithms

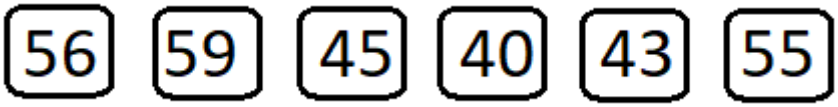
## Spring-2020 Final-Semester


## Faculty: Muhammad Adil (Asst: Prof.)

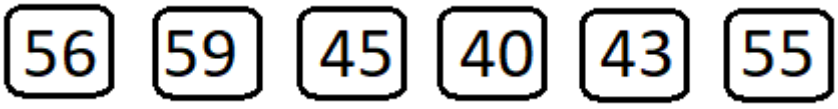
## Student Name: Muhammad Hamza Awan


**Q#1. (a) Sort the given list using Insertion Sort:**

- **56, 59, 45, 40, 43, 55**
- Algorithm of Insertion Sort is:
  - Step 1 – If it is the first element, it is already sorted. return 1;
  - Step 2 – Pick next element
  - Step 3 – Compare with all elements in the sorted sub-list
  - Step 4 – Shift all the elements in the sorted sub-list that is greater than the value to be sorted
  - Step 5 – Insert the value
  - Step 6 – Repeat until list is sorted

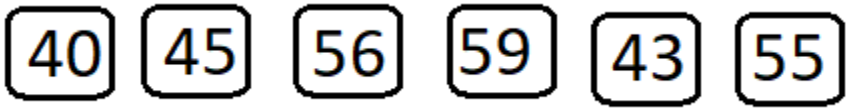



Insertion sort compares the first two elements.

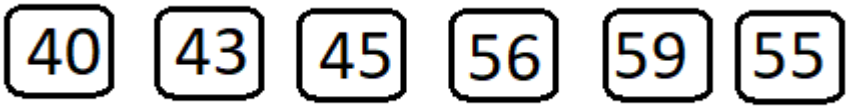



It finds that both 56 and 59 are already in ascending order. For now, 56 is in sorted sub-list. And Insertion sort moves ahead and compares 59 with 45. And finds that 45 is not in the correct position. It swaps 45 with 59. It also checks with all the elements of sorted sub-list. Here we see that the sorted sub-list has only one element, and 56 is greater than 45. So it swaps 45 with 56.

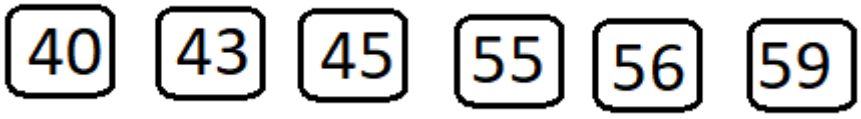

Now it compares 59 with 40. 40 is not in its place so we swap 40 with 59. Here we see that the sorted sub-list has only two elements now, and 45 and 56 is greater than 40. So it swaps 40 with 45 and 56.




Now it compares 59 with 43. 43 is not in its place so we swap 43 with 59. Here we see that the sorted sub-list has three elements now, and 40 is less than 43 while 45 and 56 are greater than 43. So it swaps 43 with 45 and 56.



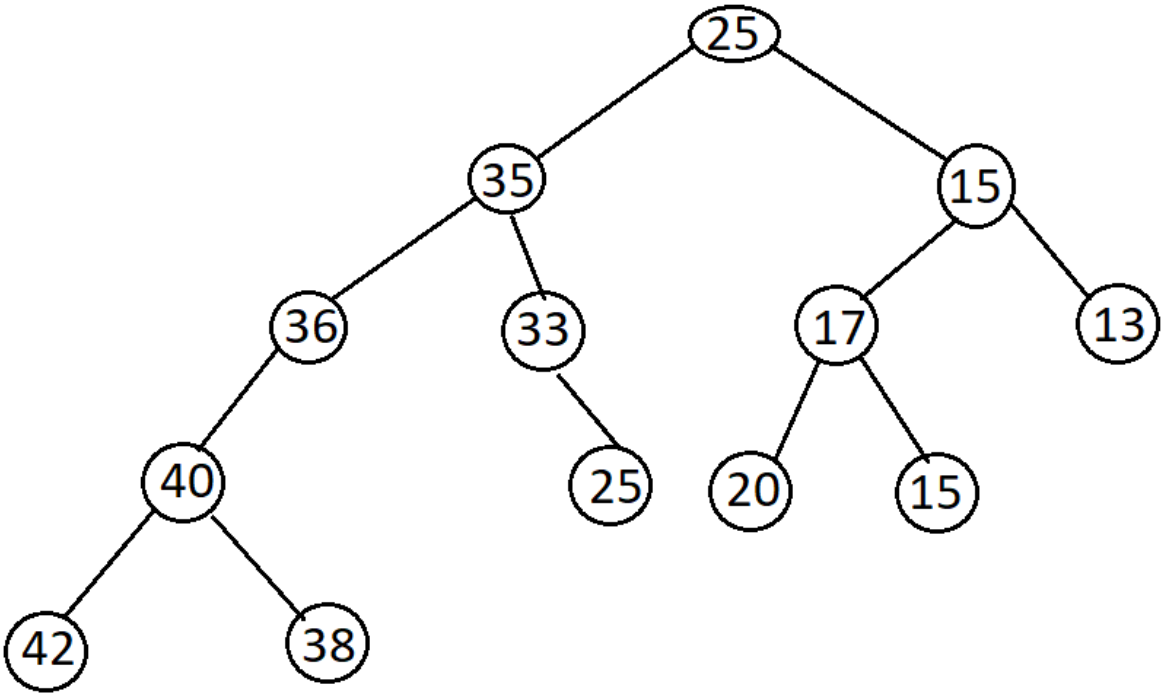

Now it compares 59 with 55. 55 is not in its place so we swap 55 with 59. Here we see that the sorted sub-list has four elements now, and 40,43 and 45 is less than 55 and 56 is greater than 55. So it swaps 55 with 56.




This process goes on until all the unsorted values are covered in a sorted sub-list.and the list will be sorted as above figure.

**Q#2. Construct Binary Trees from given list of numbers and then verify the tree.**

- **25, 15, 35, 17, 33, 36, 25, 13, 15, 40, 38, 42, 20**

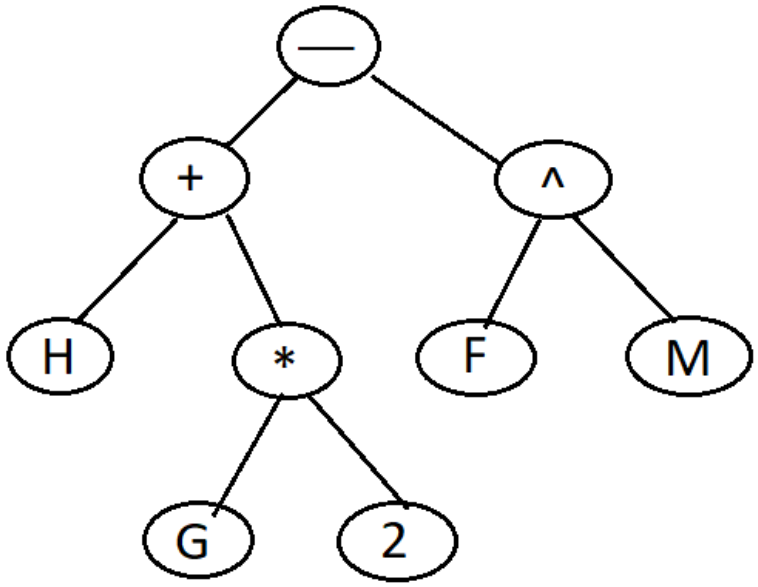



**<u>Verification:</u>**

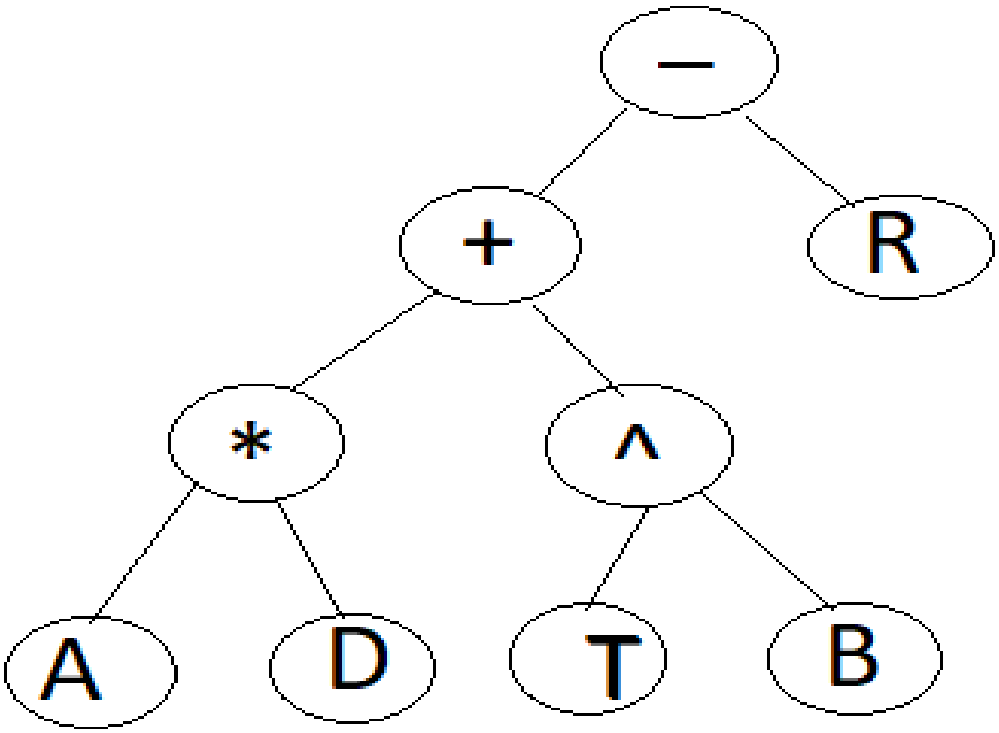
- Inorder traversal:
  - 42,40,38,36,35,33,25,25,20,17,15,15,13

**Q#3. Construct Binary Trees from given Mathematical Expressions**

i. **H + G * 2 – (F ^ M)**




ii. **A * D + T ^ B – R**



**Q#4. Apply all the three Binary Tree Traversal Techniques on each of the Tree Constructed in Q#3.**

i. **H + G * 2 – (F ^ M)**

- **In order Traversal:**

  H, +, G, *, 2, -, F, ^, M

- **Pre order traversal:**

  -, +, H, *, G, 2, ^, F, M

- **Post order traversal:**

  H, G, 2, *, +, F, M, ^, -

ii. **A * D + T ^ B – R**

- **In order Traversal:**

  A,* ,D ,+ ,T ,^ ,B ,- ,R

- **Pre order traversal:**

  -, + , * , A , D , ^ , T , B , R

- **Post order traversal:**

  A, D, *, T, B, ^, + , R, -

**Q#3. Fill in the blanks.**

i. Elements of a Tree are called **Nodes.**
ii. The graphical line drawn between Nodes of a Tree is called **edge**.
iii. Level Number of a Root is **Zero (0).**
iv. All the nodes with same Level Number belong to **same generation.**
v. The Left-Most Child Node is **Oldest Brother** Node.
vi. The Right-Most Child Node is **Youngest Brother** Node.
vii. A Tree is a **non-linear** Data Structure.
viii. An Ordered Set of Ordered Trees is called a **Forest.**