IQRA NATIONAL UNIVERSITY PESHAWAR

DEPTT. B.E. (ELECTRICAL)

8TH SEMESTER

SPRING 2020

FINAL TERM EXAMINATION

DATA STRUCTURE AND ALGORITHMS

NAME : ABDUL MATEEN

ID : 13009

INSTRUCTOR:
MUHAMMAD ADIL ASST. PROF

DATE OF SUBMISSION:
24 / JUNE / 2020

- **Attempt All Tasks.**

Q#1. (a) Sort the given list using Insertion Sort. **(10)**

56, 59, 45, 40, 43, 55

Solution:

Dry Steps:

1. For j ← 2 to n (start of outer-for-loop)
2. key ← A[j] (temporary value)
3. i ← j − 1 (counter use for while loop)
4. while (i >= 1) AND (A[i] >= key) (start of inner-while-loop)
    a. A[i + 1] ← A[i]
    b. i ← i − 1

    end of inner-while-loop
5. A[i + 1] ← key
6. Exit

Step # 1: for j ← 2 to 6

Step # 2: key ← A[j]

key ← A[2]

key ← 59

Step # 3: i ← j − 1

i ← 2 − 1

i ← 1

Step # 4: while (i >= 1) AND (A[i] >= key) (start of inner-while-loop)

(1 >= 1) //condition is true

AND

(A[1] >= 59)

(56 > = 59) //condition is false

So, while loop gets terminated

end of inner while loop

Step # 5: A[i + 1] ← key

A[1 + 1] ← key

A[2] ← 59

56, 59, 45, 40, 43, 55

Step # 1: for j ← 3

Step # 2: key ← A[j]

key ← A[3]

key ← 45

Step # 3: i ← j – 1

i ← 3 – 1

i ← 2

Step # 4: while (i >= 1) AND (A[i] >= key) (start of inner-while-loop)

(2 >= 1) //condition is true

AND

(A[2] >= 45)

(59 > = 45) //condition is true

So, while loop gets executed

a) A[i + 1] ← A[i]

A[2+1]←A[2]

A[3]←A[2]

56, 59, 59, 40, 43, 55

b) i ← i – 1

i ← 2 – 1

i ← 1

again go to check while loop condition

while (i >= 1) AND (A[i] >= key)

(1 >= 1) //condition is true

AND

(A[1] >= 45)

(56 > = 45) //condition is true

So, while loop gets executed

a) A[i + 1] ← A[i]

A[1+1]←A[1]

A[2]←A[1]

56, 56, 59, 40, 43, 55

b) $i \leftarrow i - 1$

$i \leftarrow 1 - 1$

$i \leftarrow 0$

again go to check while loop condition

while (i >= 1) AND (A[i] >= key) (start of inner-while-loop)

(0 >= 1) //condition is false

So, while loop gets terminated

end of inner while loop

Step # 5: $A[i + 1] \leftarrow key$

$A[0 + 1] \leftarrow key$

$A[1] \leftarrow 45$

45, 56, 59, 40, 43, 55

Step # 1: for $j \leftarrow 4$

Step # 2: $key \leftarrow A[j]$

$key \leftarrow A[4]$

$key \leftarrow 40$

Step # 3: $i \leftarrow j - 1$

$i \leftarrow 4 - 1$

$i \leftarrow 3$

Step # 4: while (i >= 1) AND (A[i] >= key) (start of inner-while-loop)

(3 >= 1) //condition is true

AND

(A[3] >= 40)

(59 > = 40) //condition is true

So, while loop gets executed

a) $A[i + 1] \leftarrow A[i]$

A[3+1]←A[3]

A[4]←A[3]

45, 56, 59, 59, 43, 55

b) $i \leftarrow i - 1$

$i \leftarrow 3 - 1$

$i \leftarrow 2$

again go to check while loop condition

while (i >= 1) AND (A[i] >= key)

(2 >= 1) //condition is true

AND

(A[2] >= 40)

(56 > = 40) //condition is true

So, while loop gets executed

a) $A[i + 1] \leftarrow A[i]$

A[2+1]←A[2]

A[3]←A[2]

45, 56, 56, 59, 43, 55

b) $i \leftarrow i - 1$

$i \leftarrow 2 - 1$

$i \leftarrow 1$

again go to check while loop condition

while (i >= 1) AND (A[i] >= key)

(1 >= 1) //condition is true

AND

(A[1] >= 40)

(45 > = 40) //condition is true

So, while loop gets executed

a) $A[i + 1] \leftarrow A[i]$

A[1+1]←A[1]

A[2]←A[1]

45, 45, 56, 59, 43, 55

b) $i \leftarrow i - 1$

$i \leftarrow 1 - 1$

$i \leftarrow 0$

again go to check while loop condition

while (i >= 1) AND (A[i] >= key) (start of inner-while-loop)

(0 >= 1) //condition is false

So, while loop gets terminated

end of inner while loop

Step # 5: $A[i + 1] \leftarrow key$

$A[0 + 1] \leftarrow key$

$A[1] \leftarrow 40$

40, 45, 56, 59, 43, 55

Again go to check for loop condition

Step # 1: for $j \leftarrow 5$

Step # 2: $key \leftarrow A[j]$

$key \leftarrow A[5]$

$key \leftarrow 43$

Step # 3: $i \leftarrow j - 1$

$i \leftarrow 5 - 1$

$i \leftarrow 4$

Step # 4: while (i >= 1) AND (A[i] >= key) (start of inner-while-loop)

(4 >= 1) //condition is true

AND

(A[4] >= 43)

(59 > = 43) //condition is true

So, while loop gets executed

a) $A[i + 1] \leftarrow A[i]$

$A[4+1] \leftarrow A[4]$

A[5]←A[4]

40, 45, 56, 59, 59, 55

b) $i \leftarrow i - 1$

$i \leftarrow 4 - 1$

$i \leftarrow 3$

again go to check while loop condition

while (i >= 1) AND (A[i] >= key)

(3 >= 1) //condition is true

AND

(A[3] >= 43)

(56 > = 43) //condition is true

So, while loop gets executed

a) $A[i + 1] \leftarrow A[i]$

A[3+1]←A[3]

A[4]←A[3]

40, 45, 56, 56, 59, 55

b) $i \leftarrow i - 1$

$i \leftarrow 3 - 1$

$i \leftarrow 2$

again go to check while loop condition

while (i >= 1) AND (A[i] >= key)

(2 >= 1) //condition is true

AND

(A[2] >= 43)

(45 > = 43) //condition is true

So, while loop gets executed

a) $A[i + 1] \leftarrow A[i]$

A[2+1]←A[2]

A[3]←A[2]

40, 45, 45, 56, 59, 55

b) $i \leftarrow i - 1$

$i \leftarrow 2 - 1$

$i \leftarrow 1$

again go to check while loop condition

while (i >= 1) AND (A[i] >= key)

(1 >= 1) //condition is true

AND

(A[1] >= 43)

(40 > = 40) //condition is false

So, while loop gets terminated

end of inner while loop

Step # 5: $A[i + 1] \leftarrow key$

$A[1 + 1] \leftarrow key$

$A[2] \leftarrow 43$

40, 43, 45, 56, 59, 55

Again go to check for loop condition

Step # 1: for $j \leftarrow 6$

Step # 2: $key \leftarrow A[j]$

$key \leftarrow A[6]$

$key \leftarrow 55$

Step # 3: $i \leftarrow j - 1$

$i \leftarrow 6 - 1$

$i \leftarrow 5$

Step # 4: while (i >= 1) AND (A[i] >= key) (start of inner-while-loop)

(5 >= 1) //condition is true

AND

(A[5] >= 55)

(59 > = 55) //condition is true

So, while loop gets executed

a) A[i + 1] ← A[i]

A[5+1]←A[5]

A[6]←A[5]

40, 43, 45, 56, 59, 59

b) i ← i – 1

i ← 5 – 1

i ← 4

again go to check while loop condition

while (i >= 1) AND (A[i] >= key)

(4 >= 1) //condition is true

AND

(A[4] >= 55)

(56 > = 55) //condition is true

So, while loop gets executed

a) A[i + 1] ← A[i]

A[4+1]←A[4]

A[5]←A[4]

40, 43, 45, 56, 56, 59

b) i ← i – 1

i ← 4 – 1

i ← 3

again go to check while loop condition

while (i >= 1) AND (A[i] >= key)

(3 >= 1) //condition is true

AND

(A[3] >= 55)

(45 > = 55) //condition is false

So, while loop gets terminated

end of inner while loop

Step # 5: A[i + 1] ← key

A[3 + 1] ← key

A[4] ← 55

40, 43, 45, 55, 56, 59

Again go to check for loop condition

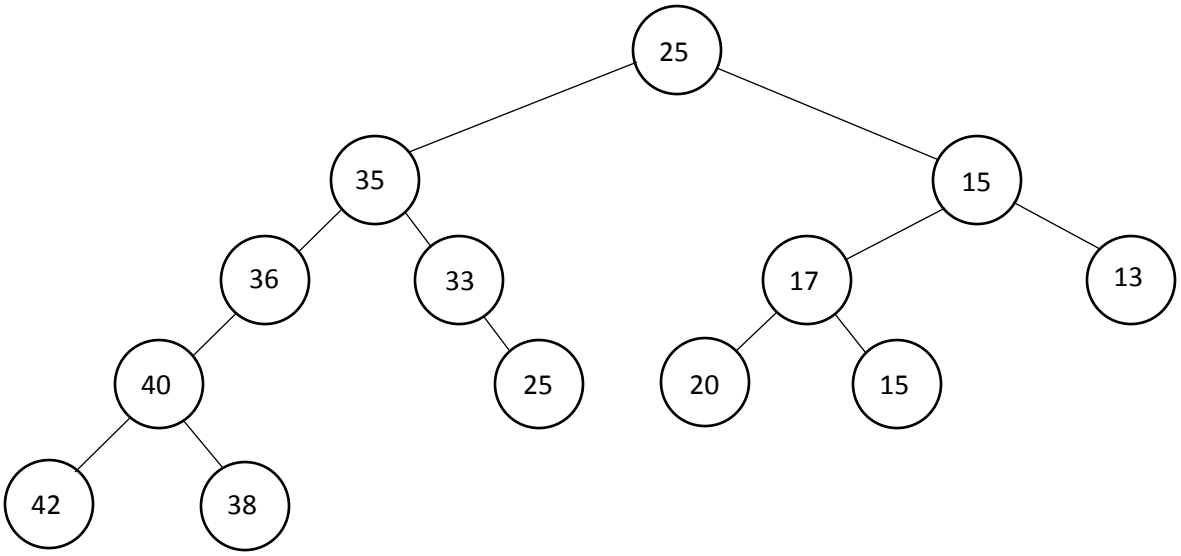
j ← 7 to 6 // condition is false

so for loop also gets terminated

program will exit here

Q#2. Construct Binary Trees from given list of numbers and then verify the tree. (10)

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

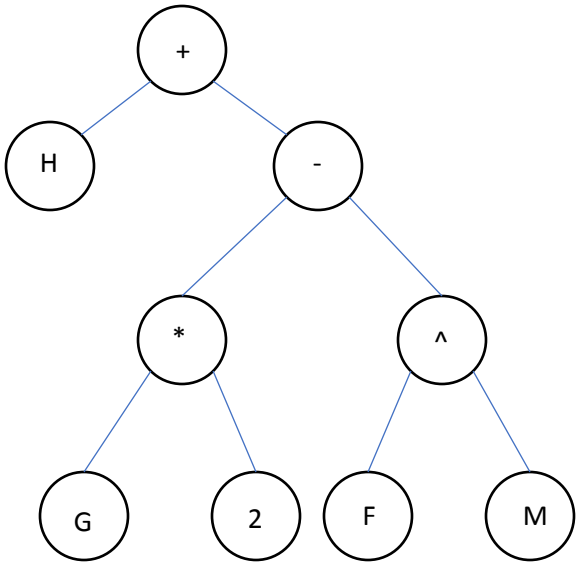


Verification using In-order-traversal:

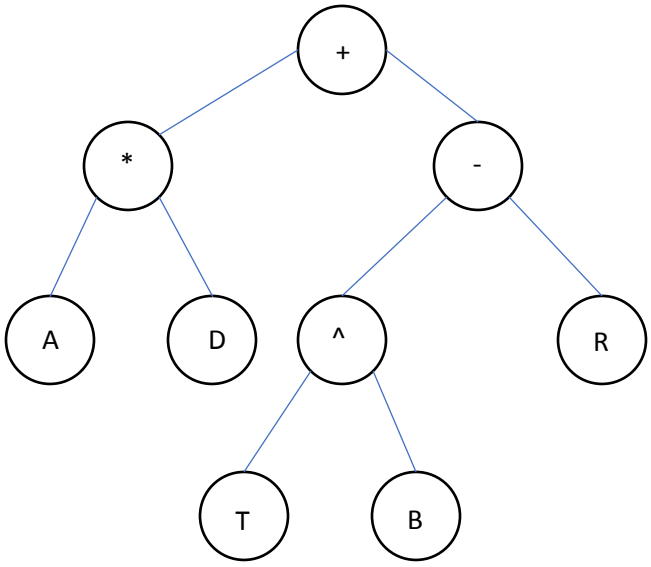
42, 40, 38, 36, 35, 33, 25, 25, 20, 17, 15, 15, 13

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

i. H + G * 2 – (F ^ M) **(05)**



ii. A * D + T ^ B – R **(05)**



Q#4. Apply all the three Binary Tree Traversal Techniques on each of the Tree constructed in Q#3. **(06 + 06)**

i. H + G * 2 – (F ^ M)

a. <u>In-Order-Traversal:</u>

- Visit left-sub tree
- Process Parent node
- Visit right-sub tree

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

b. Pre-Order-Traversal:

- Process parent node
- Visit left-sub tree
- Visit right-sub tree

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

c. Post-Order-Traversal:

- Visit left-sub tree
- Visit right-sub tree
- Process parent node

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

ii. A*D+T^B-R

a. In-Order-Traversal:

- Visit left-sub tree
- Process Parent node
- Visit right-sub tree

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

b. Pre-Order-Traversal:

- Process parent node
- Visit left-sub tree
- Visit right-sub tree

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

c. Post-Order-Traversal:

- Visit left-sub tree
- Visit right-sub tree
- Process parent node

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

Q#5. Fill in the blanks. **(08)**

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** .
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** .