Birla Institute of Technology & Science, Pilani
Work-Integrated Learning Programmes Division
Second Semester 2019-2020

Mid-Semester Test
(EC-2 Regular)

Course No.	: SS ZG519
Course Title	: DATA STRUCTURES AND ALGORITHMS DESIGN
Nature of Exam	: Closed Book
Weightage	: 35%
Duration	: 2 Hours
Date of Exam	: Saturday, 07/03/2020 (AN)

No. of Pages = 2
No. of Questions = 5

Note:
1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.

Q.1 (a) Write the pseudocode for a method that returns the average of the elements in an array *A* of *n* integers.

Note: Name your method as average. Use a while loop in case you need a looping construct.

```
average(arr, n):
    avg ← 0
    i ← 0
    while (i < n):
            avg ← avg + arr[i]
            i ← i + 1
    avg ← avg / n
    return (avg)
```

Q.1 (b) Identify the number of primitive operations for each statement in the pseudocode.

Pseudo-Code	Primitive Operation
Average(arr, n):	
avg ← 0	1(1) = 1
i ← 0	1(1) = 1
while (i < n):	(n + 1)(1) = n + 1
avg ← avg + arr[i]	n(3) = 3n
i ← i + 1	n(2) = 2n
avg ← avg / n	1(2) = 2
return (avg)	1(1) = 1

Therefore, total primitive operations = 1 + 1 + n + 1 + 3n + 2n + 2 + 1 = 6n + 6

Q.1 (c) Compute the running time of the algorithm T(*n*) and express the running time in Big-Oh notation. [3 + 3 + 2 = 8]

T(n) = 6n + 6
O(n) = n

Q.2. Solve the following recurrence equation using iterative back substitution method. $T(n) = T(n-1) + 4$ if $n \geq 1$, $T(0) = 4$ [7]

$T(n) = T(n-1) + 4$

⇨ $T(n) = T(n-1) + 4$

⇨ $T(n) = (T(n-2) + 4) + 4 = T(n-2) + 4 + 4 = T(n-2) + 4*(2)$

⇨ $T(n) = (T(n-3) + 4) + 4*(2) = T(n-3) + 4 + 4*(2) = T(n-3) + 4*(3)$

⇨ $T(n) = T(n-k) + 4k$ ---- **(A)**

Now, we know that $T(0) = 4$

Substituting, $n-k = 0$ ➔ $n = k$

Substituting **k = n** in equation **(A)**, we have,

⇨ $T(n) = T(n-k) + 4k$

⇨ $T(n) = T(n-n) + 4n$

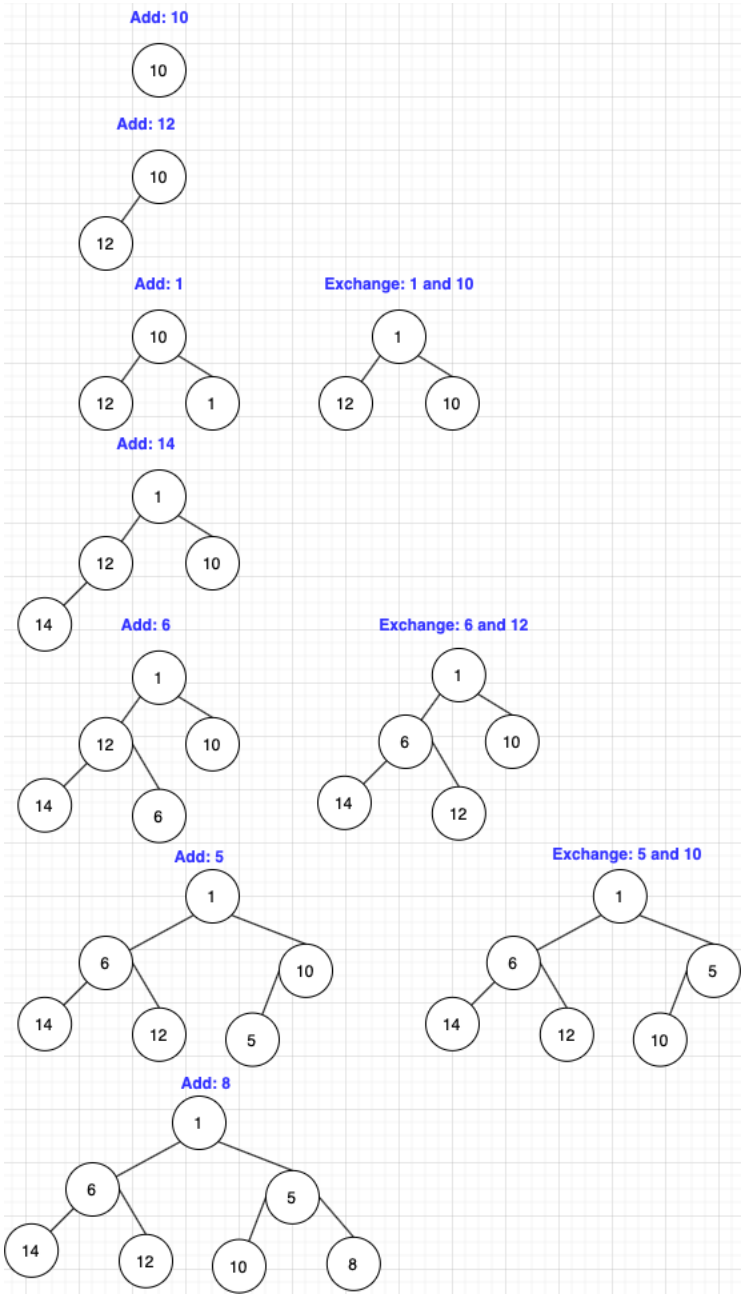
⇨ $T(n) = T(0) + 4n$

⇨ $T(n) = 4 + 4n$

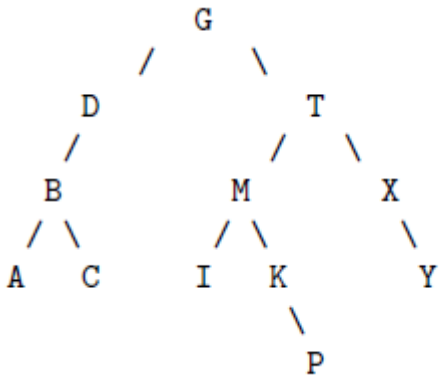
Q.3. Consider array based implementation of stack S. Assume that the maximum size of the stack is 3 elements. Show the contents of the stack after each operation along with the value of the top (t) for the following sequence of operations in the format shown in the table given below. Mention exceptions like empty/full if any for an operation. Mention 'No error' if the operation does not result in error. The initial value of top (t) is -1.
Operations: PUSH (A), POP, POP, PUSH (B), PUSH (X), PUSH (Z), PUSH (R) [7]

Operations	S[0]	S[1]	S[2]	Value of Top (t)	Error
PUSH (A)	A			0	No error
POP				-1	No error
POP				-1	EMPTY
PUSH (B)	B			0	No error
PUSH (X)	B	X		1	No error
PUSH (Z)	B	X	Z	2	No error
PUSH (R)	B	X	Z	2	FULL

Q.4. Illustrate the result of inserting the elements 10, 12, 1, 14, 6, 5 and 8 one at a time, into an initially empty binary min-heap in that order. Draw the resulting min-heap after each insertion. [7]



Q.5. For the given binary tree, write the order in which the nodes are visited for preorder, postorder and inorder traversals. [3 * 2 = 6]



PRE-Order: (ROOT -> LEFT_CHILD -> RIGHT_CHILD)
G – D – B – A – C – T – M – I – K – P – X – Y

IN-Order: (LEFT_CHILD -> ROOT -> RIGHT_CHILD)
A – B – C – D – G – I – M – K – P – T – X – Y

POST-Order: (LEFT_CHILD -> RIGHT_CHILD -> ROOT)
A – C – B – D – I – P – K – M – Y – X – T – G

****