School of Computing
Second CIA Exam – May 2023

Course Code: CSE209 **Course Name:** Data Structures & Algorithms
Duration: 90 minutes Max Marks: 50

PART A

Answer all the questions **(10 x 2 = 20)**

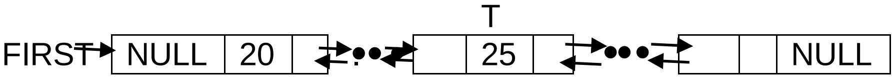
1. Write an algorithm to insert an element into beginning of a singly linked list.

```
$Algorithm\ INSERT\_AT\_BEG(first, x)$
$T = GETNODE()$
$T \rightarrow data = x$
$T \rightarrow link = first$
$first = T$
$return$
```

2. Write the algorithm to attach a new ploynomial term at the end of the polynomial which is stored as a singly linked list.

```
$Algorithm\ INSERT\_AT\_END(first, last, c, e)$
$T = GETNODE()$
$T \rightarrow data = x$
$T \rightarrow link = NULL$
$if\ first = NULL$
    $first = last = T$
$else$
    $last \rightarrow link = T$
    $last = T$
$endif$
$return$
```

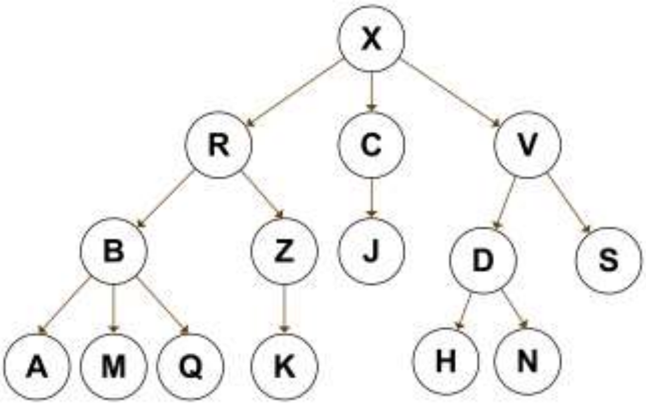
3. Let T be the address of the node to be deleted from a non-empty doubly linked list as shown below. Write the pseudocode to delete the node T.



```
$Algorithm\ DELETE\_DLLNODE(first, last, T)$
$if\ T \rightarrow next\ \neq NULL$
    $T \rightarrow next \rightarrow prev = T \rightarrow prev$
$else$
    $last = T \rightarrow prev$
    $last \rightarrow next = NULL$
$end\ if$
$if\ T \rightarrow prev\ \neq NULL$
    $T \rightarrow prev \rightarrow next = T \rightarrow next$
$else$
    $first = T \rightarrow next$
    $first \rightarrow prev = NULL$
$end\ if$
```

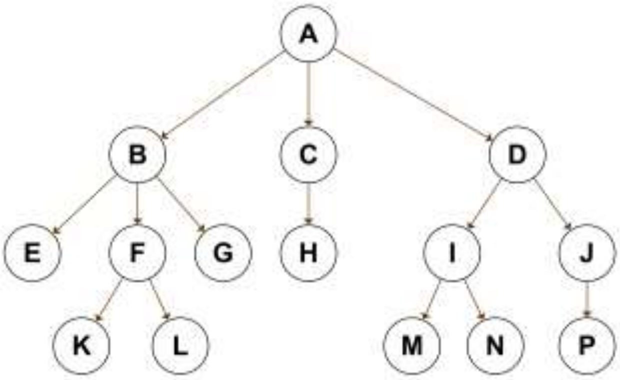
```
$RETNODE(T)$
$return$
```

4. Identify the siblings of **Q** in the following general tree.



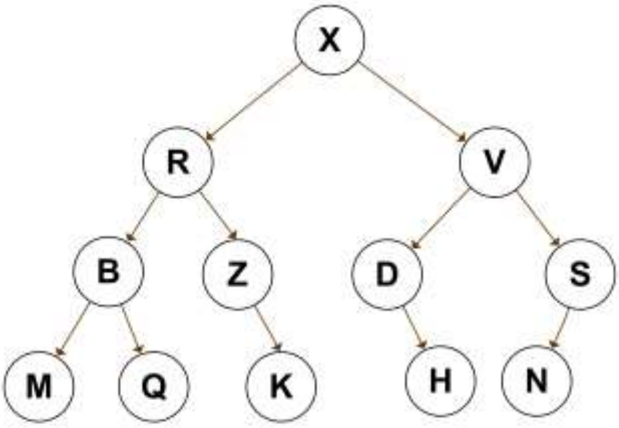
$A\ and\ M\ are\ siblings\ of\ B$

5. Write the parenthetical representation for the following general tree:



$(A\ (B(E\ F(K\ L)\ G)C(H)D(I(M\ N)J(P))))$

6. Represent the following binary tree as a sequential array.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X	R	V	B	Z	D	S	M	Q	-	K	-	H	N	-

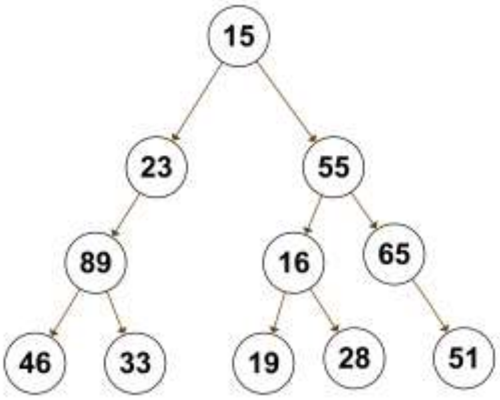
7. Define height of a binary tree.

Height of a binary tree is defined as the number of edges required to reach the farthest leaf from the root.

8. What is the maximum number of nodes in a binary tree of height h?

$Max. No. of\ nodes\ in\ a\ binary\ tree\ of\ height\ h = 2^{h+1} - 1$

9. Find the inorder traversal of the following binary tree



Inorder traversal: 46, 89, 33, 23, 15, 19, 16, 28, 55, 65, 51

10. Write the algorithm to find the maximum element in a binary search tree.

```
Algorithm MAXIMUM_BST(root)
$T = root$
while $T \rightarrow rchild \neq NULL$
    $T = T \rightarrow rchild$
end while
return $T \rightarrow data$
```

PART B

Answer any THREE questions **(3 x 10 = 30)**

11. Write the algorithm for adding two polynomials represented using singly linked list that store non-zero terms.

```
Algorithm ADD_POLY(P, Q)
$R.First = R.Last = NULL$
$t1 = P.First$
$t2 = Q.First$
while $t1 != NULL$ and $t2 != NULL$
    if $t1 \rightarrow exp > t2 \rightarrow exp$
        $R = INSERT\_AT\_LAST(R, t1 \rightarrow coef, t1 \rightarrow exp)$
        $t1 = t1 \rightarrow link$
    else if $t1 \rightarrow exp < t2 \rightarrow exp$
        $R = INSERT\_AT\_LAST(R, t2 \rightarrow coef, t2 \rightarrow exp)$
        $t2 = t2 \rightarrow link$
    else
        $coef = t1 \rightarrow coef + t2 \rightarrow coef$
```

```
            $exp = t1 \rightarrow exp$
            $if\ coef\ != 0$
                $R = INSERT\_AT\_LAST(R, coef, t2 \rightarrow exp)$
            $end\ if$
            $t1 = t1 \rightarrow link$
            $t2 = t2 \rightarrow link$
        $end\ if$
    $end\ while$
    $while\ t1\ != NULL$
        $R = INSERT\_AT\_LAST(R, t1 \rightarrow coef, t1 \rightarrow exp)$
        $t1 = t1 \rightarrow link$
    $end\ while$
    $while\ t2\ != NULL$
        $R = INSERT\_AT\_LAST(R, t2 \rightarrow coef, t2 \rightarrow exp)$
        $t2 = t2 \rightarrow link$
    $end\ while$
    $Return\ R$
```

Algorithm INSERT_AT_LAST(P, coef, exp)

```
    $n = Allocate\_Node()$
    $n \rightarrow coef = coef$
    $n \rightarrow exp = exp$
    $n \rightarrow link = NULL$
    $if\ P.First = NULL$
        $n \rightarrow link = P.First$
        $P.First = P.Last = n$
    $else$
        $P.Last \rightarrow link = n$
        $P.Last = n$
    $end\ if$
    $Return\ P$
```

12. Write the algorithms to perform insertion, deletion, and search operations in an ordered singly linked list with first pointer.

Algorithm INSERT_OSLL(FIRST, x)

```
    $T = GETNODE()$
    $T \rightarrow data = x$
    $T \rightarrow link = NULL$
    $if\ FIRST = NULL\ or\ FIRST \rightarrow data \geq x$
        $T \rightarrow link = FIRST$
        $FIRST = T$
    $else$
        $temp = FIRST$
        $while\ temp \rightarrow link\ != NULL\ \&\&\ temp \rightarrow link \rightarrow data < x$
            $temp = temp \rightarrow link$
        $end\ while$
        $T \rightarrow link = temp \rightarrow link$
        $temp \rightarrow link = T$
```

```
$endif$
$return$
```

Algorithm DELETE_OSLL(FIRST, x)

```
$if\ FIRST = NULL\ or\ FIRST \rightarrow data > x$
    $print\ "Element\ not\ present\ in\ the\ list"$
    $return$
$end\ if$
$if\ FIRST \rightarrow data = x$
    $T = FIRST$
    $FIRST = FIRST \rightarrow link$
$else$
    $prev = FIRST$
    $cur = FIRST \rightarrow link$
    $while\ cur\ != NULL\ and\ cur \rightarrow data < x$
        $prev = cur$
        $cur = cur \rightarrow link$
    $end\ while$
    $if\ cur \rightarrow data = x$
        $T = cur$
        $prev \rightarrow link = cur \rightarrow link$
    $else$
        $print\ "Element\ not\ present\ in\ the\ list"$
        $return$
    $end\ if$
$endif$
$RETNODE(T)$
$return$
```

Algorithm SEARCH_OSLL(FIRST, x)

```
$if\ FIRST = NULL\ or\ FIRST \rightarrow data > x$
    $return -1$
$end\ if$
$p = 1$
$T = FIRST$
$while\ T \neq NULL\ and\ T \rightarrow data < x$
    $p = p + 1$
    $T = T \rightarrow link$
$end\ while$
$if\ T = NULL\ or\ T \rightarrow data > x$
    $return -1$
$else$
    $return\ p$
$end\ if$
```

13. Write the algorithms to perform insertion at beginning, insertion at end, insertion at specific location into a circular doubly linked list.

Algorithm INSERT_AT_BEG_CDLL(FIRST, LAST, x)

```
$T = GETNODE()$
```

```
$T \rightarrow data = x$
$T \rightarrow prev = T \rightarrow next = NULL$
$if\ FIRST = NULL$
    $T \rightarrow prev = T \rightarrow next = T$
    $FIRST = LAST = T$
$else$
    $T \rightarrow prev = LAST$
    $T \rightarrow next = FIRST$
    $LAST \rightarrow next = T$
    $FIRST \rightarrow prev = T$
    $FIRST = T$
$return$
```

$Algorithm\ INSERT\_AT\_END\_CDLL(FIRST, LAST, x)$

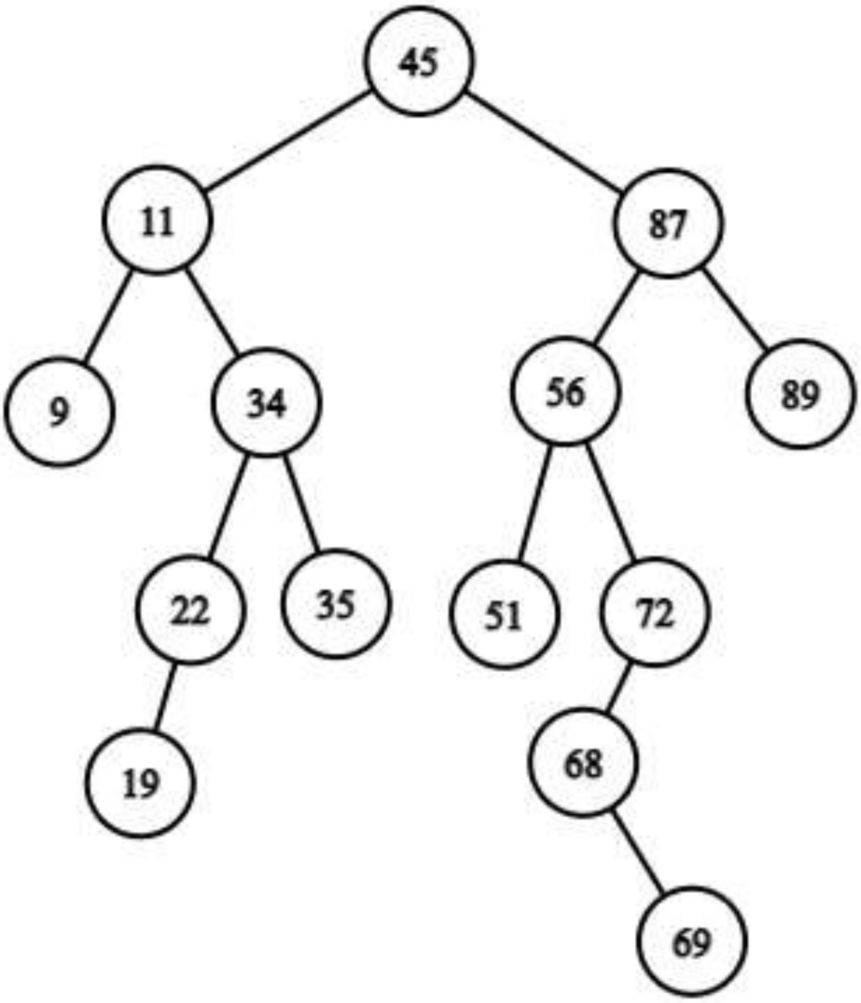
```
$T = GETNODE()$
$T \rightarrow data = x$
$T \rightarrow prev = T \rightarrow next = NULL$
$if\ FIRST = NULL$
    $T \rightarrow prev = T \rightarrow next = T$
    $FIRST = LAST = T$
$else$
    $T \rightarrow prev = LAST$
    $T \rightarrow next = FIRST$
    $LAST \rightarrow next = T$
    $FIRST \rightarrow prev = T$
    $LAST = T$
$return$
```

$Algorithm\ INSERT\_AT\_POS\_CDLL(FIRST, LAST, x, p)$

```
$T = GETNODE()$
$T \rightarrow data = x$
$T \rightarrow prev = T \rightarrow next = NULL$
$if\ FIRST = NULL$
    $T \rightarrow prev = T \rightarrow next = T$
    $FIRST = LAST = T$
    $return$
$if\ p = 1$
    $T \rightarrow prev = LAST$
    $T \rightarrow next = FIRST$
    $LAST \rightarrow next = T$
    $FIRST \rightarrow prev = T$
    $FIRST = T$
    $return$
$count = 1$
$cur = FIRST$
$while\ cur \rightarrow next\ \neq FIRST\ and\ count < p - 1$
    $count = count + 1$
    $cur = cur \rightarrow next$
$T \rightarrow prev = cur$
```

```
$T \rightarrow next = cur \rightarrow next$
$cur \rightarrow next \rightarrow prev = T$
$cur \rightarrow next = T$
$if\ LAST = cur$
    $LAST = T$
$return$
```

14. Construct a binary search tree for the following input sequence:
45, 11, 34, 87, 56, 72, 89, 51, 68, 35, 22, 19, 69, 9


45
11
87
9
34
56
89
22
35
51
72
19
68
69