School of Computing
Third CIA Exam – June 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. Find the complexity of the following algorithm:

$Algorithm\ GE(A,n)$
1. $for\ k = 1\ to\ n-1$
2. $\quad for\ i\ =\ k+1\ to\ n$
3. $\quad\quad if\ A[k,k] \neq 0$
4. $\quad\quad\quad r = A[i,k]/A[k,k]$
5. $\quad\quad\quad for\ j = k\ to\ n+1$
6. $\quad\quad\quad\quad A[i,j] = A[i,j] - r * A[k,j]$
7. $\quad\quad\quad end\ for$
8. $\quad\quad end\ if$
9. $\quad end\ for$
10. $end\ for$
11. $return$

$$T(n) = \sum_{k=1}^{n-1}\sum_{i=k+1}^{n}\sum_{j=k}^{n+1} 1 = \sum_{k=1}^{n-1}\sum_{i=k+1}^{n}(n-k+2) = \sum_{k=1}^{n-1}(n-k+2)(n-k)$$
$$= (n+1)(n-1) + \big(n(n-2)\big) + (n-1)(n-3) + \cdots + (3)(1)$$
$$= \sum_{i=1}^{n-1} i*(i+2) = \sum_{i=1}^{n-1} i^2 + 2\sum_{i=1}^{n-1} i \in O(n^3)$$

2. Trace the algorithm and find the return value when x=2 and y=5:

$Algorithm\ Compute(x,y)$
1. $if\ x\ ==\ 0$
2. $\quad return\ 0$
3. $if\ y\ ==\ 1$
4. $\quad return\ x$
5. $term\ =\ Compute(x, \lfloor y/2 \rfloor)$
6. $if\ \ y\%2\ ==\ 0$
7. $\quad return\ term\ *\ term$
8. $else$
9. $\quad return\ term\ *\ term\ *\ x$

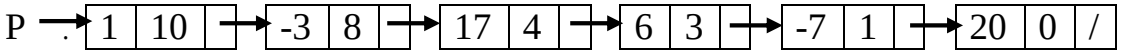
Ans. 32

3. Define Ω-Notation.
It is an asymptotic notation to represent the lower bound of a function. For algorithms, it specifies the best case time complexity.

4. Evaluate the following postfix expression when a=7, b=18, c=3, d=10 **using stack**: $abc/+d*$. Write the contents of stack at each step of evaluation.
   Stack Contents:

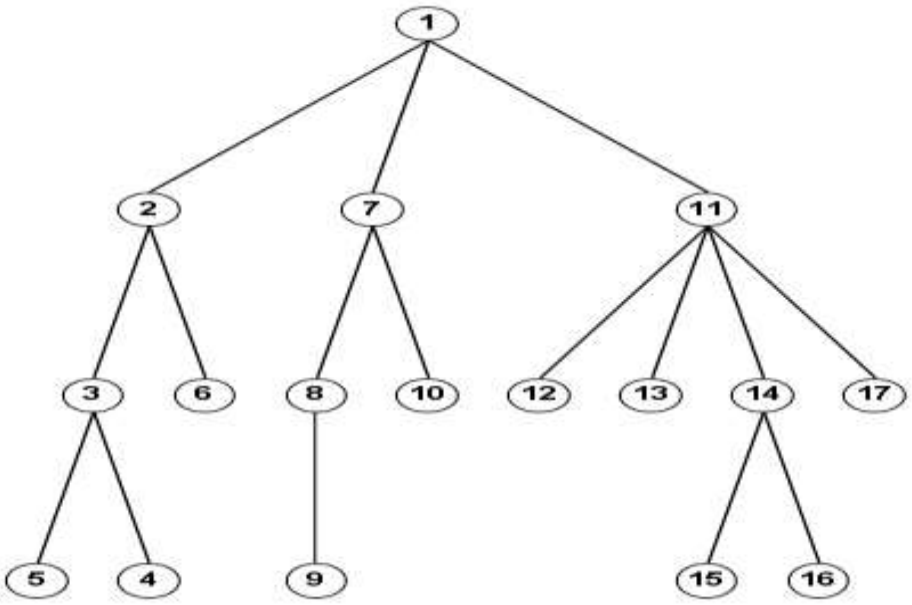
a	b	c	/	+	d	*
Push 7	Push 18	Push 3	Divide	Add	Push 10	Multiply
		3				
	18	18	6		10	
7	7	7	7	13	13	130

5. Represent the following polynomial using singly linked list:
   $P = x^{10} - 3x^8 + 17x^4 + 6x^3 - 7x + 20$

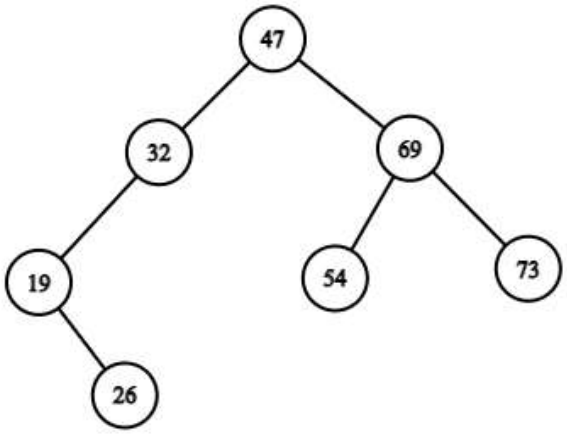


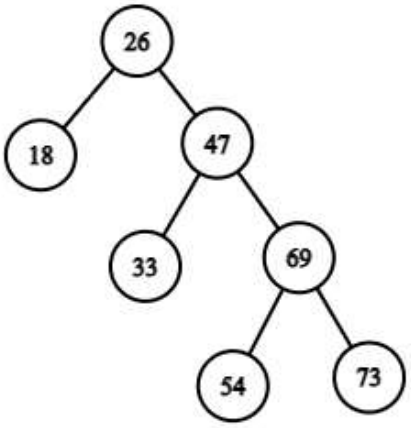
6. Write an algorithm to search for the position of a given element x in a singly linked list
   $Algorithm\ SEARCH\_SLL(FIRST, x)$
   1. $p = 1$
   2. $T = FIRST$
   3. $while\ T \neq NULL\ and\ T \rightarrow data \neq x$
   4. $\quad p = p + 1$
   5. $\quad T = T \rightarrow link$
   6. $end\ while$
   7. $if\ T = NULL$
   8. $\quad return - 1$
   9. $else$
   10. $\quad return\ p$
   11. $end\ if$

7. Draw the general tree whose parenthetical representation is:
   (1 (2 (3 (4 5) 6) 7 (8 (9) 10) 11 (12 13 14 (15 16) 17)))

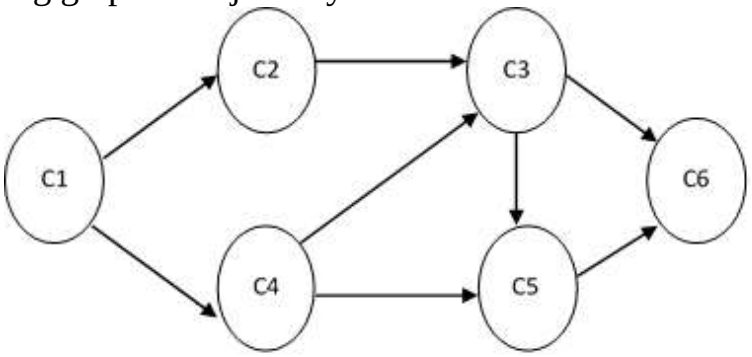


8. Search for 26 in the following splay tree and draw the resultant tree after splaying.





9. Represent the following graph as adjacency matrix.



Adjacency Matrix:

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	1	0	0	0
3	0	0	0	0	1	1
4	0	0	1	0	1	0
5	0	0	0	0	0	1
6	0	0	0	0	0	0

10. Perform first three iterations of selection sort on the following input sequence:
15, 12, 25, 17, 20, 9, 11, 8, 10, 4

Pass 1: Swap 15 and 4

4	12	25	17	20	9	11	8	10	15

Pass 2: Swap 12 and 8

4	8	25	17	20	9	11	12	10	15

Pass 3: Swap 25 and 9

4	8	9	17	20	25	11	12	10	15

PART B

Answer any TWO questions **(2 x 10 = 20)**

11. Write the algorithm for converting infix expression into postfix using stack.

$Algorithm\ Infix\_to\_Postfix(Infix)$

1. $createStack(S)$
2. $j = 1$
3. $for\ i = 1\ to\ length(Infix)$
4. $\quad x = Infix[i]$
5. $\quad if\ x\ is\ operand$
6. $\quad\quad Postfix[j] = x$
7. $\quad\quad j = j + 1$
8. $\quad else\ if\ x\ is\ '('$
9. $\quad\quad Push(S, top, x)$
10. $\quad else\ if\ x\ is\ ')'$
11. $\quad\quad while\ Peek(S, top) \neq '('$
12. $\quad\quad\quad t = Pop(S, top)$
13. $\quad\quad\quad Postfix[j] = t$
14. $\quad\quad\quad j = j + 1$
15. $\quad\quad end\ while$
16. $\quad\quad Pop(S, top)$
17. $\quad else$
18. $\quad\quad while\ Priority\big(Peek(S, top)\big) \geq Priority(x)$
19. $\quad\quad\quad t = Pop(S, top)$
20. $\quad\quad\quad Postfix[j] = t$
21. $\quad\quad\quad j = j + 1$
22. $\quad\quad end\ while$
23. $\quad\quad Push(S, top, x)$
24. $\quad end\ if$
25. $\quad while\ not\ isEmpty(S, top)$
26. $\quad\quad\quad t = Pop(S, top)$
27. $\quad\quad\quad Postfix[j] = t$
28. $\quad\quad\quad j = j + 1$
29. $\quad end\ while$
30. $\quad Postfix[j] = '\backslash 0'$
31. $\quad return\ Postfix$

12. Write the algorithms to perform insertion, deletion, and search operations in an ordered doubly linked list.

$\boldsymbol{Algorithm\ INSERT\_ODLL(FIRST, LAST, x)}$

1. $T = GETNODE()$
2. $T \rightarrow data = x$
3. $T \rightarrow prev = T \rightarrow next = NULL$
4. $temp = FIRST$
5. $while\ temp \neq NULL\ and\ temp \rightarrow data < x$
6. $\quad\quad temp = temp \rightarrow next$

7. $end\ while$
8. $T \rightarrow next = temp$
9. $if\ temp = NULL$ $//\ insert\ at\ end$
10. $T \rightarrow prev = LAST$
11. $LAST \rightarrow next = T$
12. $LAST = T$
13. $else$
14. $T \rightarrow prev = temp \rightarrow prev$
15. $if\ temp \rightarrow prev = NULL$ $//\ Insert\ at\ begining$
16. $FIRST = T$
17. $else$
18. $temp \rightarrow prev \rightarrow next = T$
19. $endif$
20. $temp \rightarrow prev = T$
21. $end\ if$
22. $return$

Algorithm DELETE_ODLL(FIRST, LAST, x)

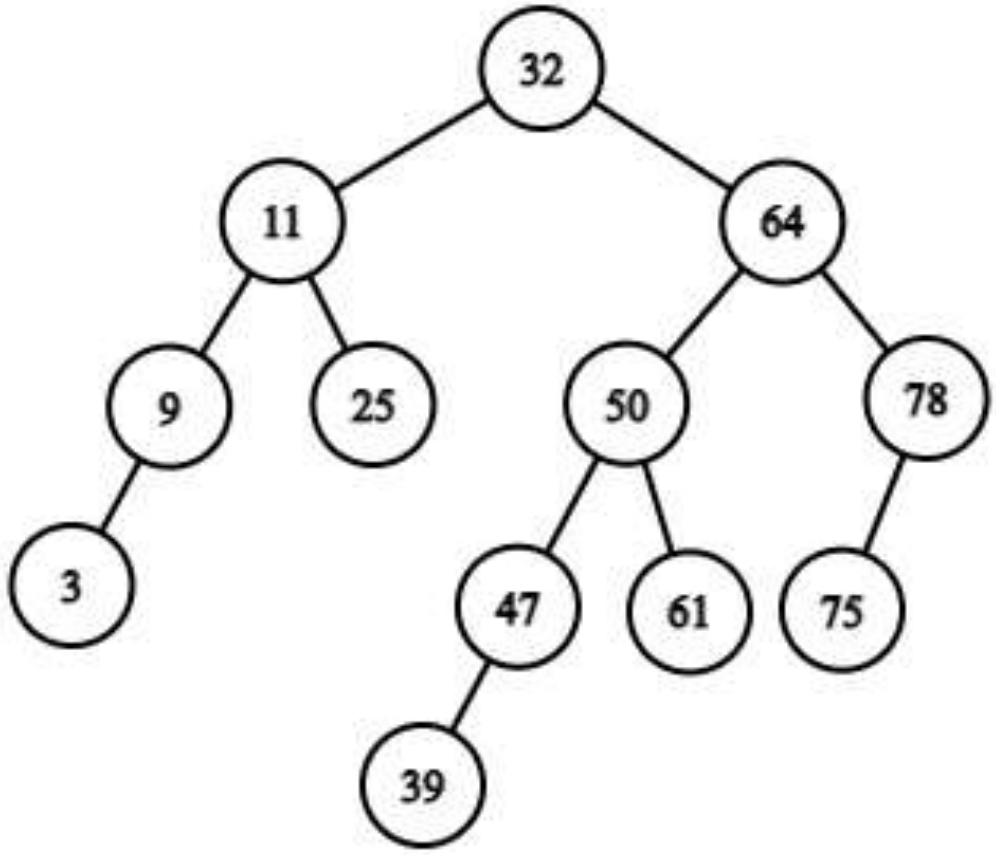
1. $temp = FIRST$
2. $while\ temp \neq NULL\ and\ temp \rightarrow data < x$
3. $temp = temp \rightarrow next$
4. $end\ while$
5. $if\ temp = NULL\ or\ temp \rightarrow data > x$
6. $print\ "Element\ not\ present\ in\ the\ list"$
7. $return$
8. $end\ if$
9. $if\ temp \rightarrow prev \neq NULL$
10. $temp \rightarrow prev \rightarrow next = temp \rightarrow next$
11. $else$ $//delete\ at\ begining$
12. $FIRST = FIRST \rightarrow next$
13. $end\ if$
14. $if\ temp \rightarrow next \neq NULL$
15. $temp \rightarrow next \rightarrow prev = temp \rightarrow prev$
16. $else$ $//delete\ at\ end$
17. $LAST = LAST \rightarrow prev$
18. $end\ if$
19. $RETNODE(temp)$
20. $return$

Algorithm SEARCH_ODLL(FIRST, LAST, x)

1. $p = 1$
2. $T = FIRST$
3. $while\ T \neq NULL\ and\ T \rightarrow data < x$
4. $p = p + 1$
5. $T = T \rightarrow next$
6. $end\ while$
7. $if\ T = NULL\ or\ T \rightarrow data > x$
8. $return - 1$
9. $else$
10. $return\ p$
11. $end\ if$

13. Construct an AVL tree for the following input sequence: 25, 32, 64, 11, 78, 50, 9, 3, 61, 75, 47, 39

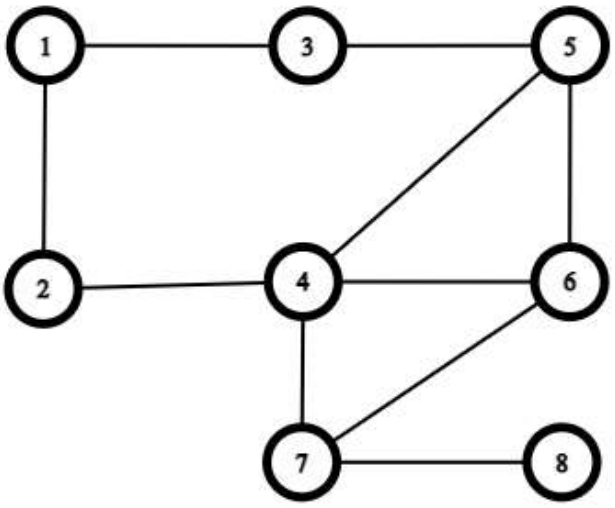
Final Tree



PART – C

Answer all the questions **(1 x 10 = 10)**

14.(i) Represent the following graph as adjacency list and perform breadth first traversal. (6 Marks)



Breadth First Search:

Name	dist	Visited	Parent	adjacency list
1	∞ 0	∞ 1	-1	→ 2 → 3
2	∞ 1	∞ 1	∞ 1	→ 1 → 4
3	∞ 1	∞ 1	∞ 1	→ 1 → 5
4	∞ 2	∞ 1	∞ 2	→ 2 → 5 → 6 → 7
5	∞ 2	∞ 1	∞ 3	→ 3 → 4 → 6
6	∞ 3	∞ 1	∞ 4	→ 4 → 5 → 7
7	∞ 3	∞ 1	∞ 4	→ 4 → 6 → 8
8	∞ 4	∞ 1	∞ 7	→ 7

Let S = 1

Queue

1	2	3	4	5	6	7	

u = 1 v = 2, 3

2.dist = 1.dist + 1 = 0 + 1 = 1 2.parent = 1

3.dist = 1.dist + 1 = 0 + 1 = 1 3.parent = 1

u = 2 v = ~~1~~, 4

4.dist = 2.dist + 1 = 1 + 1 = 2 4.parent = 2

u = 3 v = ~~1~~, 5

5.dist = 3.dist + 1 = 1 + 1 = 2 5.parent = 3

u = 4 v = ~~2~~, ~~5~~, 6, 7

6.dist = 4.dist + 1 = 2 + 1 = 3 6.parent = 4

7.dist = 4.dist + 1 = 2 + 1 = 3 7.parent = 4

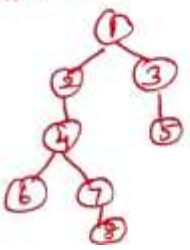
u = 5 v = ~~3~~, ~~4~~, ~~6~~

u = 6 v = ~~4~~, ~~7~~, ~~7~~

u = 7 v = ~~4~~, ~~6~~, 8

8.dist = 7.dist + 1 = 3 + 1 = 4 8.parent = 7

Breadth First Tree



(ii) Sort for the following input sequence using heap sort: 5, 41, 34, 7, 56, 12, 81, 56, 60
(4 Marks)