SET A – ANSWER KEY

SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2023-24 (ODD)

Test: CLA-CT-1 **Date: 04-09-2023**
Course Code & Title: 21CSC201J Data Structures and Algorithms **Duration:** 1 hour 40 min
Year & Sem: II Year / III Sem **Max. Marks:** 50

Course Articulation Matrix:

Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Program Specific Outcomes		
													PSO-1	PSO-2	PSO-3
CO1	1	2	3	-	-	-	-	-	-	-	-	3	3	-	-
CO2	2	3	3	-	-	-	-	-	-	-	-	3	3	-	-
CO3	2	3	3	-	-	-	-	-	-	-	-	3	3	-	-
CO4	2	3	3	-	-	-	-	-	-	-	-	3	3	-	-
CO5	3	2	3	-	-	-	-	-	-	-	-	3	3	-	-

Part - A
(10 x 1 = 10 Marks)

Instructions: Answer all

Q. No	Question	Marks	BL	CO	PO	PI Code
1	Which function has the maximum asymptotic complexity? a) f1(n)=n^(3/2) b) f2(n)=n^(log n) c) f3(n)=n log n d) f4(n)=2^n	1	L1	1	2	2.5.3
2	What does the following declaration mean? int (*ptr) [10]; a) ptr is an array of 10 integers b) ptr is a pointer to an array of 10 integers c) ptr is an array of pointers of 10 integers d) array of 10 integers	1	L1	1	2	2.5.2

3	What will be the output of the following C code?
	#include<stdio.h>
	void main()
	{
	char *s="hello";
	char *p=s;
	printf("%c\t%c", p[0], s[1]);
	}
	
	a) h e
	b) l l
	c) l o
	d) l e
		1	L2	1	2	2.5.2

4	What will be the output of the following code?	1	L2	1	2	2.5.2

	int main() {
	struct fruit
	{
	int apple;
	int mango;
	}F1,*F2;
	F1.apple=1000;
	F1.mango=20;
	F2=&F1;
	printf("%d ",F2.mango);
	}
	
	a) 1000
	b) 20
	c) Compile time error
	d) 100					

5	We can certainly get a conclusion about an algorithm's ___________ scenario from asymptotic analysis. a) best case b) best case, average case, and worst case c) worst case d) average case	**1**	**L1**	**1**	**2**	**2.5.1**
6	Which of the following is a practical example of a doubly linked list? a) A browser cookie file. b) A quest in a game that lets users retry stages. c) A game in which the player runs forward. d) A first-in-first out scheduling system.	**1**	**L1**	**2**	**2**	**2.5.1**
7	The matrix contains m rows and n columns. The matrix is called sparse matrix if _________ a) Total number of Zero elements > (m*n)/2 b) Total number of Zero elements = m + n c) Total number of Zero elements < m/n d) Total number of Zero elements = m-n	**1**	**L2**	**2**	**2**	**2.5.1**
8	Which type of linked list stores the address of the head node in the next pointer of the last node? a) Singly Linked List b) Doubly Linked List c) Hashed List d) Circular Linked List	**1**	**L1**	**2**	**2**	**2.5.1**
9	Which of the following information is stored in a doubly-linked list's nodes? a) Value of node b) Address of next node c) Address of the previous node d) All of the above	**1**	**L2**	**2**	**1**	**1.7.1**

10	Identify the correct option for creating a node in linked list at first time
	a) ptr = (struct node)malloc(sizeof(struct node *));
	b) ptr = (struct node *)calloc(size(struct node *));
	c) ptr = (struct node *)malloc(sizeof(struct node *));
	d) ptr = (struct node *)calloc (size(struct node));
		1	**L2**	**2**	**1**	**1.7.1**

Part – B
(4 x 5 = 20 Marks)

Instructions: Answer all

11	Define a structure called cricket which contains the following data members: player_name, team_name and batting_average. Using cricket, create structure array which holds 50 players. Write a C program to read and display the information about all the 50 players.	**5**	**L3**	**1**	**2**	**2.5.2**

ANSWER:

```
#include <stdio.h>
#include <string.h>

// Define the cricket structure
struct cricket {
    char player_name[50];
    char team_name[50];
    float batting_average;
};

int main() {
    // Create an array of cricket structures to hold 50 players
    struct cricket players[50];

    // Read information about 50 players
    for (int i = 0; i < 50; i++) {
        printf("Enter details for Player %d:\n", i + 1);

        printf("Player Name: ");
        scanf("%s", players[i].player_name);

        printf("Team Name: ");
        scanf("%s", players[i].team_name);

        printf("Batting Average: ");
        scanf("%f", &players[i].batting_average);
    }

    // Display information about all 50 players
    printf("\nPlayer Information:\n");
    for (int i = 0; i < 50; i++) {
        printf("Player %d:\n", i + 1);
        printf("Name: %s\n", players[i].player_name);
        printf("Team: %s\n", players[i].team_name);
        printf("Batting Average: %.2f\n\n",
players[i].batting_average);
    }

    return 0;
}
```

12	Arrange the growth rate of 2^n ,$\sqrt{n}$, n^2, 1, log n, n log n, 3^n and n in increasing order of growth and explain the upper, lower and average bound asymptotic	**5**	**L3**	**1**	**2**	**2.5.3**

notations.

ANSWER:

1.Upper Bound (Big O Notation, O):

- Big O notation provides an upper bound on the growth rate of a function. It represents an asymptotic upper limit.
- For example, if we say that a function f(n) is O(g(n)), it means that there exists a constant c and a value n0 such that for all n >= n0, f(n) <= c * g(n).
- It is used to describe the worst-case time complexity of an algorithm.

2.Lower Bound (Omega Notation, Ω):

- Omega notation provides a lower bound on the growth rate of a function.
- If we say that a function f(n) is Ω(g(n)), it means that there exists a constant c and a value n0 such that for all n >= n0, f(n) >= c * g(n).
- It is used to describe the best-case time complexity of an algorithm.

3.Average Bound (Theta Notation, Θ):

- Theta notation provides both upper and lower bounds on the growth rate of a function, effectively bounding the function from above and below.
- If we say that a function f(n) is Θ(g(n)), it means that there exist constants c1, c2, and n0 such that for all n >= n0, c1 * g(n) <= f(n) <= c2 * g(n).
- It is used to describe the average-case time complexity of an algorithm when the upper and lower bounds are the same.

Let's arrange the functions in increasing order of growth rate:

❖ 1 (constant) - This function does not depend on n, so it grows the slowest.
❖ log n (logarithmic) - Logarithmic growth is slower than constant but faster than any polynomial growth.
❖ √n (square root) - The square root grows slower than logarithmic but faster than linear.
❖ n (linear) - Linear growth is slower than square root but faster than any polynomial growth with a higher degree.
❖ n log n (linearithmic) - Linearithmic growth is faster than linear but slower than quadratic.
❖ 2n (exponential) - Exponential growth is much faster than any polynomial growth.
❖ 3n (exponential) - Similar to 2n, it's exponential and grows faster than any polynomial.
❖ n^2 (quadratic) - Quadratic growth is slower than exponential but faster than linearithmic.

So, the functions arranged in increasing order of growth rate are:

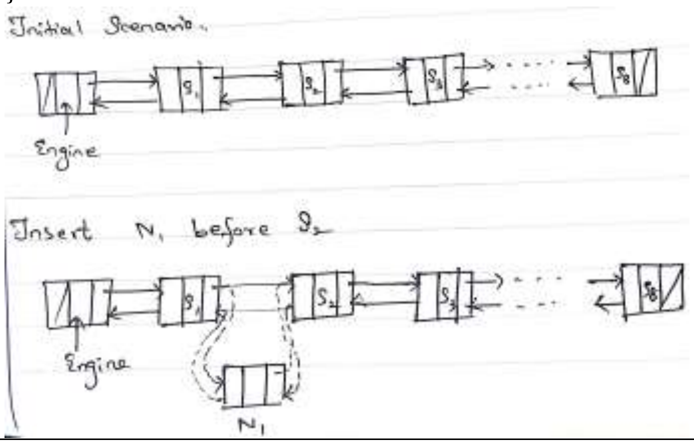
1 (constant)
log n (logarithmic)
√n (square root)
n (linear)
n log n (linearithmic)
n^2 (quadratic)
2n (exponential)
3n (exponential)

13	Consider the following scenario, Engine – S1 – S2 – S3- S4 -S5 -S6 S7-S8 Here, S1 to S8 are the compartments that are connected with forward and backward links. Write an algorithm to insert the compartment named N1, before the compartment "S2" and display the compartment position from the engine.	**5**	**L3**	**2**	**2**	**2.5.2**

INSERT

```
void insert (int N1 ,int L ,position P)
{
position Newnode;
Newnode =(struct Node *)malloc(sizeof(struct Node)) ;
if (Newnode ==NULL)
        printf("Fatal Error");
else
{
        Newnode ->Element =N1;
        P = L -> Next;
        While (p!=NULL && P -> Element !=S2)
              P =P -> Next;

        Newnode ->Next =P ->Next;
        Newnode ->Prev =P;
        P ->Next =Newnode;
        Newnode→Next→Prev = P;
}
}
```



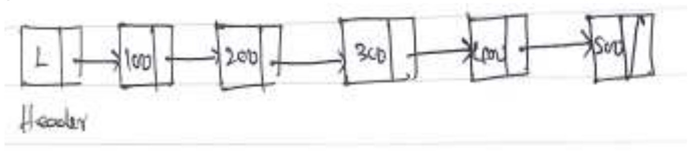
14	Implement Singly Linked List to insert and display the following elements 100, 200, 300, 400 and 500	**5**	**L3**	**2**	**2**	**2.5.2**

ANSWER:

```
LIST createlist()
{
        LIST L;
        L=(struct Node *)malloc(size of(struct    Node));
        if(L==NULL)
                printf("fatal error");
        else
        {
                L->data=-1;
                L->Next=NULL;
        }
        return L;
}
```

INSERT

```
void insert (int X ,int L ,position P)
{
```

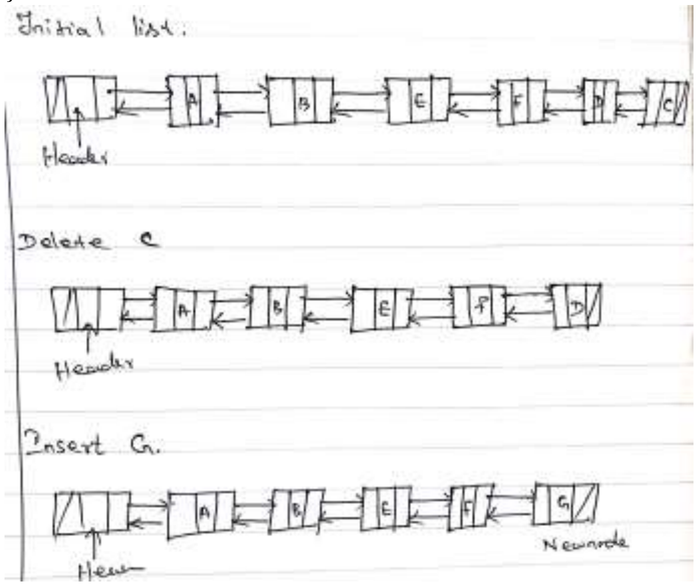
	position Newnode;
	Newnode =(struct Node *)malloc(sizeof(struct Node)) ;
	if (Newnode ==NULL)
	        printf(“Fatal Error”);
	else
	{
	While(P→Next!=NULL)
	          P=P→Next;
	    Newnode ->Element =X;
	    Newnode ->Next =NULL;
	    P ->Next =Newnode;
	}
	}
						

Part – C **(2 x 10 = 20 Marks)**						
15	a) Find the lower bound of the function $f(n) = n^4 + 3n^3 + 2n+1$; $g(n) = \Omega(n^3+4)$. **(5 marks)** Answer: To find lower bound of f(n), we have to find c and n0 such that $0 \le c.g(n) \le f(n)$ for all $n \ge n0$ $0 \le c \times g(n) \le f(n)$ $0 \le c \times g(n) \le n^4 + 3n^3 + 2n+1$ $0 \le 1.n^3+4 \le n^4 + 3n^3 + 2n+1 \rightarrow$ true, for all $n \ge 1$ $0 \le n^3 \le n^4 + 3n^3 + 2n+1 \rightarrow$ true, for all $n \ge 1$ Above both inequalities are true and there exists such infinite inequalities. So, $f(n) = \Omega(g(n)) = \Omega(n^3)$ for c = 1, n0 = 1 b) Find the upper bound of the function $f(n) = 3n+2$; $g(n)= O(n)$. **(5 marks)** Answer: To find upper bound of f(n), we have to find c and n0 such that $0 \le f(n) \le c \times g(n)$ for all $n \ge n0$ $0 \le f(n) \le c \times g(n)$ $0 \le 3n+2 \le c \times g(n)$ $0 \le 3n+2 \le 3n+2n$, for all $n \ge 1$ (There can be such infinite possibilities) $0 \le 3n+2 \le 5n$	**10**	**L3**	**1**	**2**	**2.5.3**

	So, c = 5 and g (n) = n, n0 = 1					
	(or)					

16	Write a C program is to display the elements and calculate the sum of n numbers using dynamic memory allocation.
	Sample Input:
	Enter the number of elements : 5
	Enter the elements :
	23
	34
	12
	34
	56
	Output: The sum of elements is 159
	
	ANSWER:
	#include <stdio.h>
	#include <stdlib.h>
	int main()
	{
	
	int i;
	int count;
	int *arr;
	int sum = 0;
	printf("Enter the total number of elements you want to enter : ");
	scanf("%d", &count);
	        arr = (int *)malloc(count * sizeof(int));
	for (i = 0; i < count; i++)
	{
	printf("Enter element %d : ", (i + 1));
	scanf("%d", arr + i);
	sum += *(arr + i);
	}
	printf("sum is %d \n", sum);
	free(arr);
	return 0;
	}
	OUTPUT:
	Enter the total number of elements you want to enter : 5
	Enter element 1 : 1
	Enter element 2 : 2
	Enter element 3 : 3
	Enter element 4 : 4
	Enter element 5 : 4
	sum is 14
		10	**L3**	**1**	**2**	**2.5.2**

17	Arun prepared the list of items required for his college tour. The items are Item A, Item B, Item C, Item D, Item E and Item F. Later on, he realized that, the last Item C from the list is no more required and he removed the item from the list and he wanted to include a new item named as Item G at the end of the list. Use the double linked list data structure and write the proper pseudocode along with visual representation. **<u>ANSWER:</u>** **<u>DELETION: - 5 marks</u>** Void Delete(int C, List L) {	**10**	**L3**	**2**	**2**	**2.5.2**

	Position P, Temp;
	P=P→Next;
	While(P!=NULL && P→Element!=C)
	        P = P→Next;
	If(P→Next == NULL)
	{
	    P→Prev→Next==NULL;
	    Free(P)
	}
	INSERTION: - 5 marks
	void insert (int G ,int L)
	{
	position Newnode;
	Newnode =(struct Node*)malloc(sizeof(struct Node)) ;
	if (Newnode ==NULL)
	        printf("Fatal Error");
	else
	{
	Newnode ->Element =G;
	
	While(P→Next!=NULL)
	            P=P→Next;
	
	      P->Next= Newnode;
	      Newnode->Prev=P
	}
	}
	 Initial list:
	A
	B
	E
	F
	D
	C
	Header
	Delete C
	A
	B
	E
	F
	D
	Header
	Insert G.
	A
	B
	E
	F
	G
	Newnode
	Head					

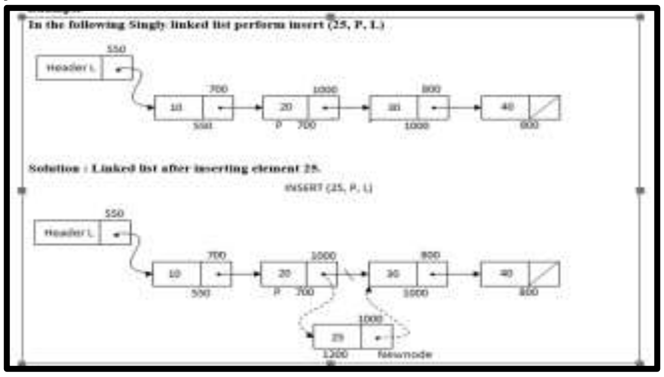
	(or)					

18	Vande Bharat Express initially has an engine as a header. Later compartments are linked with engines one by one in series. The compartments can be attached and detached with the engine or in between the compartments. Give a visual representation and implement a data structure for attaching and detaching a compartment in between.
	
	Answer:
	CREATE LIST ------------------------ 2 MARKS
	LIST createlist()
	{
	        LIST L;
	        L=(struct Node *)malloc(size of(struct Node));
		10	**L3**	**2**	**2**	**2.5.2**

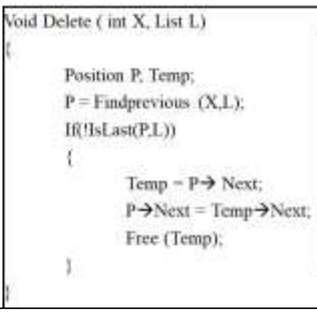
```
            if(L==NULL)
                        printf("fatal error");
            else
            {
                        L->data=-1;
                        L->Next=NULL;
            }
            return L;
}
```

INSERT -- ---------------------- 4 MARKS

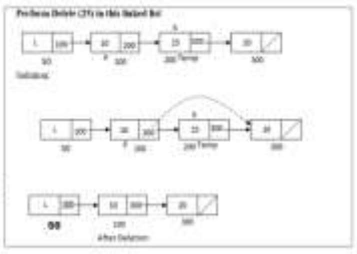
```
void insert (int X ,int L ,position P)
{
position Newnode;
Newnode =(struct Node *)malloc(sizeof(struct Node)) ;
if (Newnode ==NULL)
            printf("Fatal Error");
else
{
            Newnode ->Element =X;
            Newnode ->Next =P ->Next;
            P ->Next =Newnode;
}
}
```


In the following Singly linked list perform insert (25, P, L)
550
Header L
700
1000
800
10
20
30
40
550
P 700
1000
800
Solution : Linked list after inserting element 25.
INSERT (25, P, L)
1000
25
1200
Newnode

DELETE ------------------------ 4 MARKS

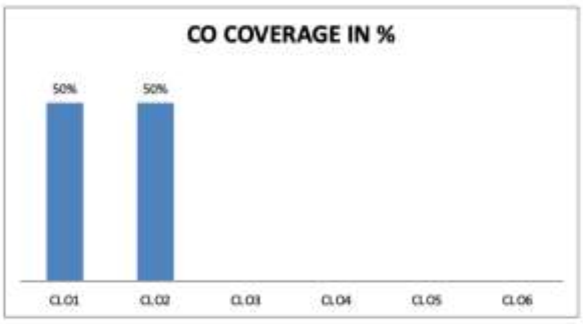


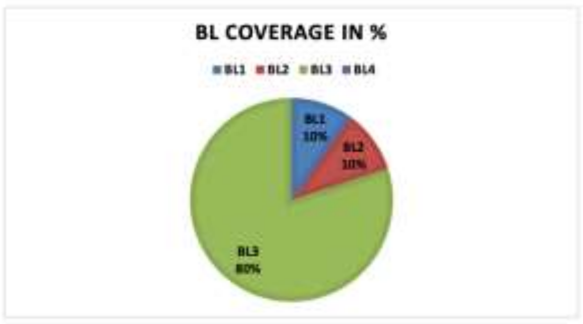
```
Void Delete ( int X, List L)
{
        Position P, Temp;
        P = Findprevious (X,L);
        If(!IsLast(P,L))
        {
                Temp = P→ Next;
                P→Next = Temp→Next;
                Free (Temp);
        }
}
```



***Program Indicators are available separately for Computer Science and Engineering in AICTE examination reforms policy.**

Course Outcome (CO) and Bloom's level (BL) Coverage in Questions





Approved by the Audit Professor/Course Coordinator