CO-201

Date : / /
Page No.

1[a] How do you measure the complexity of an Algorithm Find out the complexity for the followings:
(i) Linear search (Best, Average & Worst case)
(ii) Binary search (Worst case)
(iii) Bubble sort (Worst case) 3

i) Linear Seach

Best Case : when element is present on the first index

| 10 | 20 | 30 | 40 | ---- key = 10

∴ Time Complexity → O(1)

B) Explain average case time complexity of linear search. [3 marks]

Average Case : when element is present on the mid part

| 10 | 20 | 30 | 40 | 50 | key = 30

∴ Time complexity = $\Theta(n/2) = \Theta(n)$

Worst Case : when the element is present at the last index of an array of n elements.
n comparisons of TC = 1 are req
∴ Time Complexity → O (n)

ii) Binary Search : Here, in every step, length of the array is halved
Consider no. of operations by the following table

| no. of comparisons | Length of array(n) |
|---|---|
| 1 | n/2 |
| 2 | $n/2^2$ |
| 3 | $n/2^3$ |
| ⋮ | ⋮ |
| k | $1 = n/2^k$ |

thus, $1 = n/2^k$
$\therefore k = \log_2(n)$

thus, T.C. = $O(\log_2(n))$

iii) Bubble Sort : The bubble sort uses 2 loops
for all elements {
for elements bigger than i
swap
}

O-(. $n^2$)

Date : / /
Page No.

Define complexity of an algorithm. Distinguish between time and space complexity? 1

Complexity of an algorithm is a measure of the amount of time and/or space required by an algorithm for an input of a given size.

| | TC | SC |
|---|---|---|
| calculates | time needed | calculates memory needed by program |
| counts | time for all statements | counts memory space for all input I/p |
| worst case | TC is considered | memory allocated for all variables is considered |

Date : / /
Page No.

(b) What is Recursion? Which data structure is used to perform recursion? Explain with the help of an example. 1

Recursion is defined as the process of refering a something in terms of itself.
In programming, it refers to the process of a function calling itself.

Stack data structure is used for recursion.

Eg)
```
void Recurse()
{
    recurse();
}
```

b) Write a recursive function to find the greatest common divisor(GCD) of two numbers (5)

```
int GCD (int n1, int n2)
{
    if (n2 != 0)
        return hcf (n2, n1 % n2);
    else
        return n1;
}
```

Date : / /
Page No.

(d) What is the time complexity of the function?

```
int fun(int n)
{
 int count = 0;
 for (int i = n; i > 0; i /= 2)
   for (int j = 0; j < i; j++)
     count += 1;
 return count;
}
```

→ Work $\log_2 n$ times (similar to binary)
→ $kn$ times (when $k<1$)

$\log_2 n \times (kn)$

$O : (n(\log_2 n))$

(b) Find out the time and space complexity of following C- program

```
int i , j, k, sum=0;
for (i=0;i<n; i++)  1
  for(j=0;j<n; j++)  2
    for(k=0;k<9999;k++) 3
      sum++;
```

① takes $O(n)$ time (independent)
② takes $O(m)$ time (independent)
③ takes constant time $O(1)$

$\therefore TC \rightarrow O(m \times n)$ | $SC \rightarrow O(1)$ as sum is only used.

Q2. Define and explain –

- Big -Oh notation
- Big-omega notation

i) Big O is a mathematical notation that describes the limiting behaviour of a function when argument tends to $\infty$.

$\therefore$ if $f(n) \leq c.g(n) \; \forall \; n \geq n_0$ and $c > 0$ then $f(n) \rightarrow O(g(n))$ where worst case of $f(n)$ is $g(n)$.

It is used to calc worst case time and space complexity.

Date : / /
Page No.

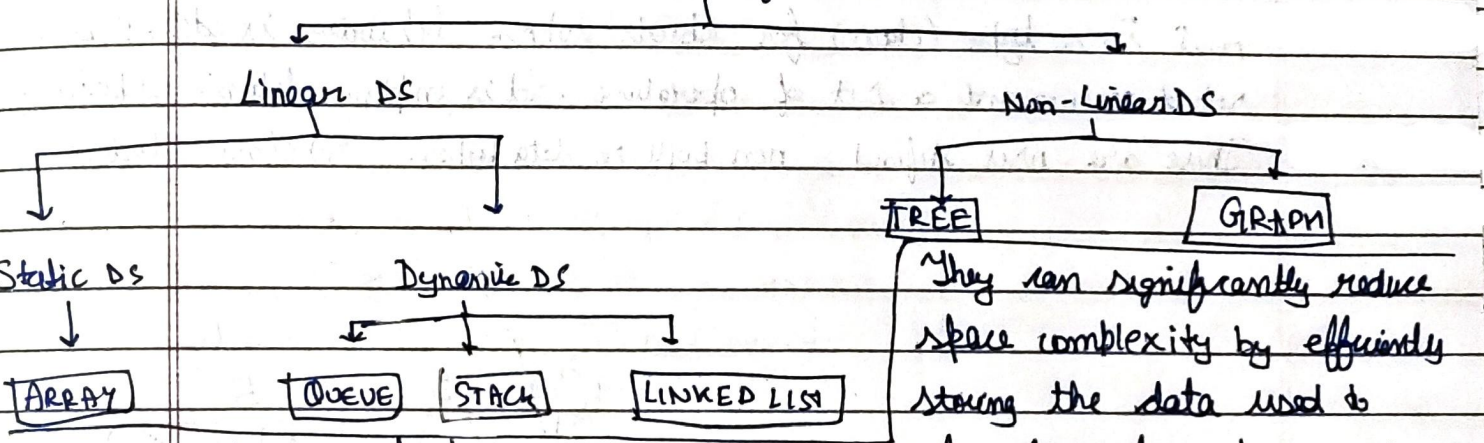
ii) Big Omega ($\Omega$) is a mathematical denotation defining the lower bound of an algorithm.

$\therefore 0 \leq c\,g(n) \leq f(n)$ for all $n \geq n_0$

$\therefore \Omega(g(n))$ gives best case of $f(n)$

a) Classify different types of Data structures. Also discuss how they can help in reducing space complexity for different real world problems. (3)

Data structures are classified as.




They can significantly reduce space complexity by efficiently storing the data used to solve a problem. Eg:) Storing the friendlist of a person in the form of Graph (DS) is more efficient than creating a simple list.

What is Data Structure? Distinguish between Primitive and non primitive data structure.

A data structure is a way of organising data that considers not only the items stored, but also their relationship to each other

| PRIMITIVE | NON - PRIMITIVE |
|---|---|
| It is a fundamental data type which allows only one type of data to be stored | It is a user defined DS which stores the data of diff types in single entity |
| Eg:) int, bool. | Eg:) Tree, Queues |

Date : / /
Page No.

b) Give the sequence of argument values that result when the following program is invoked for each of integers 3, 4 and 5. (4)

```
int puzzle(int n)
{
    If (n==1)
        return 1;
    If (n%2==0)
        return puzzle(n/2);
    else
        return puzzle(3*n+1);
}
```

n=3 | 3 10 5 16 8 4 2 1

n=4 | 4 2 1

n=5 | 5 16 8 4 2 1

[b] Explain the following terms with suitable example
(i) Abstract Data Type (ADT)

ADT is a type (class) for objects whose behaviour is defined by a set of values and a set of operations. It is implementation-independent. These are user defined, non built in data types. E.g.) Class Student.

2)

a) What is the output of the following code snippet?

```
#include <stdio.h>
int main()
{
    int arr[][3]={{1,2},{3,4,5},{5}};
    printf("%d%d%d", sizeof(arr), arr[0][2], arr[1][2]);
    return 0;
}
```

i) ii) iii)

OUTPUT : 36 0 5

array is

i) Sizeof = 3×3×4 = 36

ii)iii) in diagram

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 2 | ii) 0 |
| 1 | 3 | 4 | iii) 5 |
| 2 | 5 | 0 | 0 |

Consider a two-dimensional array A [20][10]. Assume 4 words per memory cell, the base address of array A is 100, elements are stored in row-major order and first element is A[0][0]. What is the address of A[11][5] ?

[b] A 2D Array DATA [6][8] is stored in a column major order with base address 301 and size 1 calculate the address of DATA [2][4]. 2

COLUMN MAJOR ORDER

array → DATA [m][n]

element → DATA [i][j]

size → S

base address → B

∴ add ([i][j]) = ((j*m)+i)xS+B

= 301 + ((4x6)+2)x1

= 301 + (26)

= 327

ROW MAJOR ORDER

array → DATA [m][n]

element → DATA [i][j]

size → S

base → B

∴ add ([i][j]) = B + (i*n+j)xS

= 100 + (11*10+5)x4

= 100 + (115)x4

= 560

indexing starts from 1

Date : / /
Page No.

(a) Let A be a two-dimensional array declared as follows:
A: array[1...10] [1...15] of integer;
Assuming that each integer takes one memory location. The array is sorted in row-major order and the first element of the array is stored at location 100, what is the address of the element A[i][j] ?

A

Given Base add = 100
m = 10
n = 15

for row-major order

$\therefore$ add (A[i][j]) = B.A. + ((i-1) x n + (j-1)) x 1

= 100 + (15i - 14 + j - 1)

add = 15i + j + 84

Q1 A) An m*n matrix is said to have a saddle point if some entry a[i][j] is the smallest value in row I and largest value in column j. Write an algorithm that determines the location of a saddle point if one exists. [3 marks]

1. Create a 1D array corresponding to each row
2. Find the minimum element of the current row and store the column index of the minimum element.
3. Check if the row minimum element is also maximum in its column. We use the stored column index here
4. If yes, then Saddle point else continue till the end of matrix.

5. Write a program that grades multiple-choice test. Suppose there are eight students and ten questions, and the answers marked by students are stored in a two dimensional array. Each row records a student's answers to the questions, as shown in the following array:

Students' Answers to the Questions:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Student 0 | A | B | A | C | C | D | E | E | A | D |
| Student 1 | D | B | A | B | C | A | E | E | A | D |
| Student 2 | E | D | D | A | C | B | E | E | A | D |
| Student 3 | C | B | A | E | D | C | E | E | A | D |
| Student 4 | A | B | D | C | C | D | E | E | A | D |
| Student 5 | B | B | E | C | C | D | E | E | A | D |
| Student 6 | B | B | A | C | C | D | E | E | A | D |
| Student 7 | E | B | E | C | C | D | E | E | A | D |

Key to the Questions:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Key | D | B | D | C | C | D | A | E | A | D |

Remember sequence of questions are not same for every student. Sequence of student-0 and key are matching i.e. <0,1,2,..,9>. For student-1, sequence of question numbers are <1,2, 3,..,9,0> (means Q.no 1 appears as Q.no. 0), for student-2, sequence of questions are <2,3,4,..,9,0,1> and so on. Program should print number of correct answers for every student. (8 Marks)

```
class GradeExam {
char student_ans[][] = { ≡ }
char key[] = { D, B, D, C ——— }
};
for (int i=0 ; i < student_ans.length ; i++)
{ int correct = 0;
  for (int j=0; j < key.length; j++)
  {
    if (key[j] == student_ans[i][j])
    { correct++; }
  }
cout << "Student" << i << " correct count is " << correct << endl;
```

Each element of an array A[20][50] requires 4 byte of storage where first element is stored at location 2000. Determine the location of A[10][10] when array is stored as

i. Row major

ii. Column major

Date : / /

Page No.

| m = 20 | s = 4 |
|---|---|
| n = 50 | B.A = 2000 |

i.) for Row major

add (X[i][j]) = B.A. + (i x n + j) x s

= 2000 + (10 x 50 + 10) x 4

= 2000 + (510) x 4 = 4040

ii.) for Column Major

add (X[i][j]) = B.A + (i + j x m) x s

= 2000 + (10 + 10 x 20) x 4

= 2000 + (10 + 200) x 4

= 2840

(a) Suppose you are given an array s[1...n] and a procedure reverse(s,i,j) which reverses the order of elements in between positions i an j (both inclusive). What does the following sequence do, where 1≤k≤n:

```
reverse(s,1,k);
reverse(s,k+1,n);
reverse(s,1,n);
```

Explain with the help of an example.

The sequence of steps rotated S array left by K position

Eg) S = [1 2 3 4 5 6 7] and n = 7, k = 2

∴ i.) reverse (S, 1, 2) : S = [2, 1, 3, 4, 5, 6, 7]

ii.) reverse (S, 3, 7) : S = [2, 1, 7, 6, 5, 4, 3]

iii.) reverse (S, 1, 7) : S = [3, 4, 5, 6, 7, 1, 2] ✓

(b) Write a recursive function to count the number of occurrences of the number 24 in an array of n integers.

OR

```
int count (int array[], int n)
{ int no = 24;
  if (n>0) { if (array[n-1] = no)
    { count++;
      length--;
      count (array, length);
      return count;
    }
  else
  return count;
```

(i) What does the following fragment of C-program pr

```
char c[] = "IAH 2019";
char *p = c;
printf("%s", p + p[3] - p[1]);
```

p[3] (ascii) 2019

p+4 = 2019

4th index

Date : / /
Page No.

b) Implement a stack using arrays in C++

```
class Stack {
    private:
        int total;
        int *arr;
        int tos;
    public:
        Stack (int cap) {
            total = cap;
            arr = new int [total];
            tos = -1;
        }
        Void push (int item) {
            if (isFull()) {
                cout << "Full";
                return
            }
            tos++
            arr [tos] = item;
        }
```

(i) Deletion operation of STACKS

```
        void pop() {
            if (isEmpty()) {
                cout<< "Empty";
            }
            int temp = arr [tos];
            arr [tos] = 0
            tos --
            return temp;
        }
        int peek() {
            if (isEmpty()) { ——— }
            return arr [tos];
        }
        Int size() { return tos + 1; }
        bool isEmpty { return size() == total }
        void display() {
            for (int i = tos; i >= 0; i--) {
                cout << arr [i] << endl;
            }
        }
}
```

Date : / /
Page No.

) Give an algorithm to convert an infix expression to a prefix expression using Stacks and also illustrate with an example. (4)

1. Reverse the infix expression. Note that while reversing swap ( and )

Write an algorithm to convert infix expression to reverse police notation using underlying Data Strcuture STACKS. Also illustrate the steps for the following expression:
$(A - (B * C))/(D + (E \hat{} F))$

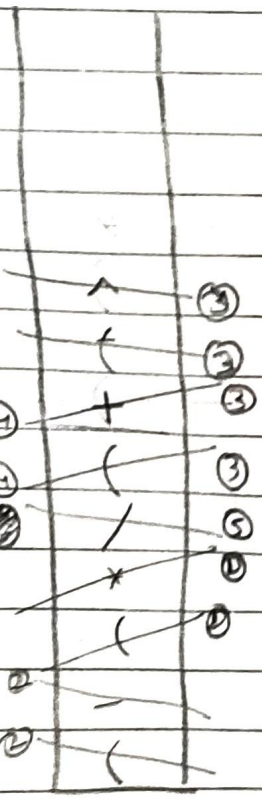
2. Obtain the postfix expression of the reversed string
   i.) If scanned char is operand then print it
   ii.) If it is an operator ;
      a) If precedence is greater than tos then push it
      b) If lower than pop until its precedence is greater than tos and then push
      c) associativity case.
   iii.) if it is ( push in stack.
   iv) if it is ) pop until ) is found.

3. Reverse the obtained String → Ans. ✓.

(A-(B*C)) / (D+ (E^F))

o/p

ABC*-DEF^+/




Translate the following infix expression into postfix expression:
A + (B * C – (D / E / F) * G) * H

= A+ (B* C - ( DE/F) G *) *H

= A+( BC* (DE/F) G * -) * H *

= A B C * D E / F / G * - H * +

(b) Convert the given infix expression into its equivalent Prefix and Postfix expression:
((A+B)^C)*(D-E)^(F/G))

POST
= (AB+C^) * (DE-)^(FG/)
= AB+C^ (DE-) (FG/)^*
= AB+C^DE-FG/^*

PRE
= (^+ABC) * (-DE) ^ (/FG)
= * ^ + ABC ^ - DE / FG

Date : / /
Page No.

(c) The result of evaluating the postfix expression 10, 5, +, 60, 6, /, *, 8, – is

10 5 + 60 6 / * 8 -

= 142

| 142 | 150-8 |
|---|---|
| 8 | - |
| 150 | 15×10 |
| 10 | 60/6 |
| 6 | |
| 60 | 10+5 |
| 15 | 10+5 |
| 5 | |
| 10 | |

5. Let S be a stack of size n ≥ 1. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and pop operation take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation.

(a) For m ≥ 1, define the stack-life of m as the time elapsed from the end of Push(m) to the start of the pop operation that removes m from S.

(b) Compute the average stack-life of an element of this stack.

(6 marks)

a) Stack life for $i^{th}$ element;

No of pushes : $(n-i)$

No of pops : $(n-i)$

$\therefore$ time → $\underbrace{2(n-i)x}_{\text{pops push}} + (2(n-i)+1)y$

Stack life = $2(n-i)(x+y)+y$

| n |
|---|
| ⋮ |
| i+1 |
| i |
| ⋮ |
| 3 |
| 2 |
| 1 |

b) Averge Stack life

$$\frac{\sum_{i=1}^{n} 2(n-i)(x+y)+y}{n} = \frac{2(x+y)\sum_{i=1}^{n}(n-i)+y}{n}$$

$$= \frac{2(x+y)\,n(n-1)}{n \times 2} + y = \boxed{n(x+y)-x}$$

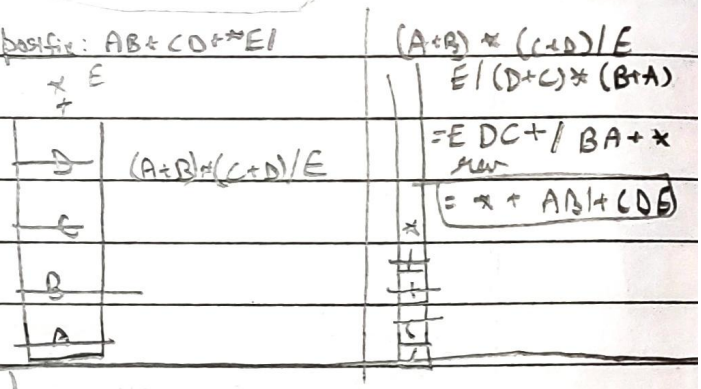
3. Write algorithm to determine if the parentheses (or other brackets) in a string are balanced and properly nested. For example, the string "((())())()" contains properly nested pairs of parentheses, but the string ")()(" does not; and the string "())" does not contain properly matching parentheses. Give an algorithm that returns 'true' (or 1) if a string contains properly nested and balanced parentheses, and 'false' (or 0) if otherwise. Do not write code for any data structure used like linked list/stack/queue. (6 marks)

4. Use a stack to test for balanced parentheses, when scanning the following expressions. Only consider the parentheses [,],(,),{,} . Ignore the variables an operators. Example inputs (valid balanced) are:
(a) [a+{b/(c-d)+e/(f+g)}-h]
(b) [a{b+[c(d+e)-f]+g)
Write algorithm to test if given input string contains balanced parentheses.

```
bool Balanced ( String expr){
    Stack<char> temp
    for (int i=0; i < expr.length ,i++){
        if (temp. empty ()
        {
        temp. push (expr [i]);
        }
    else if ((temp. top () == '(' && expr[i] == ')')
            || ——— { ——— }
            || ——— [ ——— ])
        {
        temp. pop();
        }
    else
        {
        temp. push (expr [i]);
        }
    if (temp. empty ())
        {
        return true;
        }
    return false;
    }
```

(a) Perform the following operations using stack:
i) Postfix to Prefix expression: AB+CD+*E/
ii) Evaluate the following Postfix expression: 2 4 6 * - 9 8 3 / + * 5 ^ 2 +
Where ^ is an exponential.

i) postfix: AB+CD+*E/




Give an algorithm to convert a Prefix expression to an Infix expression using Stacks. (5)

1. Read the prefix expression in reverse order (right to left)
2. If symbol is an operand, push it into the stack.
3. If symbol is an operator (□), pop two operands (op1 and op2) and perform ans = op1 □ op2 (concatenate), and push ans in stack.
4. Repeat steps (2-3) until string is traversed
5. pop the last element of Stack as answer.

op2
op1

Date : / /
Page No.

ii)

| |
|---|
| |
| |
| |
| |
| |
| 3 |
| 8 |
| 9 |
| -22 |
| ~~24~~ |
| ~~6~~ |
| ~~4~~ |
| ~~2~~ |

Create a data structure *twoStacks* that represents two stacks. A single array A[1...MAXSIZE] is used to implement *twoStacks*. The two stacks S1 and S2 grow from opposite ends of the array. Variables top1 and top2 (top1<top2) point to the location of the topmost element in each of the stacks. Following functions must be supported by *twoStacks*.

Push1(int x)→ pushes x to first stack.
Push2(int x)→ pushes x to second stack.
Pop1()→ pops an element from first stack and return the popped element.
Pop2()→ pops an element from second stack and return the popped element.
Write an algorithm for the implementation of above said *twoStack*. (5

S

```
class twoStacks { int *a, len, top1, top2;
public:
    twoStacks (int n) {
        len = n;
        a = new int [n];
        top1 = -1
        top2 = len ! }
```

Write the algorithm for the implementation of above said *twoStack*. If the space is to be used efficiently, what should be the condition for "stack full"? [6+2=8]

```
void push1(int x)
{
    if (top1 < top2 - 1) {
        top1++;
        a[top1] = x;
    }
    else { cout << "overflow" ; }
}
int pop1 ( ) {
    if (top1 >= 0) {
        int x = a[top1];
        top1--;
        return x;
    }
    else { cout << "underflow" ; }
```

```
void push2(int x)
{
    if (top1 < top2 - 1) {
        top2--;
        a[top2] = x;
    }
    else { cout << "overflow" ; }
}
int pop2 ( ) {
    if (top2 < len) {
        int x = a[top2];
        top2++;
        return x;
    }
    else { cout << "underflow" ; }
```

Date : / /

Q3. A) Suppose the following stack of integers is in memory where STACK is allocated N=6 memory cells:

TOP =2 STACK: 5, 2, 3, ___, ___, ___

Find the output of following program segment:

1. Call POP(STACK, ITEMA)
2. Call POP(STACK, ITEMB)
3. Call PUSH(STACK, ITEMB + 2)
4. Call PUSH(STACK, 8)
5. Call PUSH(STACK, ITEMA + ITEMB)
6. Repeat while TOP ≠ -1:
   Call POP(STACK, ITEM)
   Write: ITEM.
   [End of loop]

1.) ITEMA = 3
TOP = 1
STACK = 5, 2, -, -, -, -

2.) ITEMB = 2
TOP = 0
STACK = 5, 2, -, -, -, -

3.) TOP = 1
STACK = 5, 4, -, -, -, -

4. TOP = 2
STACK = 5, 4, 8, -, -, -

5. TOP = 3
STACK = 5, 4, 8, 5, -, -

6. ITEM = 5
STACK = -, -, -, -, -, -

a) Write a Program that will create a new stack containing the elements 31, 51, 21, 71, 91 in ascending order. Minimize the number of intermediary stacks used. (3)

```
Stack<int> SortStack(Stack<int> &input){
    stack<int> tempStack;
    while (!input.empty()){
        int tmp= input.top();
        input.pop();
        while (!tmpstack.empty() && tmpStack.top()>tmp)
        { input.push(tmpStack.pop());
          tmpStack.pop();
        }
        tmpStack.push(tmp);
    }
    return tmpStack);}
int main(){
    // input 31, 51, 21, 71, 91
```

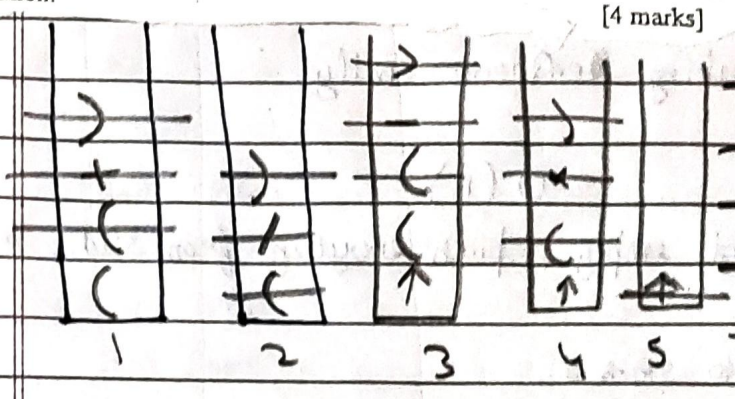
Q2. A) Compute the postfix equivalent of the following infix arithmetic expression

$$((5+7)/6)\uparrow((4-1)*2)$$

Where ↑ represents exponentiation. Assume normal operator precedence. Also evaluate the generated postfix expression using stack. Show status of stack after each execution.

[4 marks]

Date : / /
Page No.

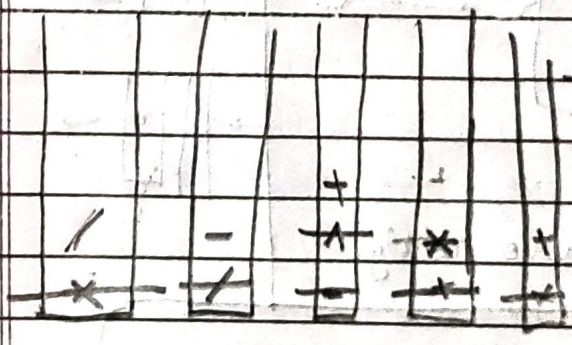



1. 57
2. 57+6/
3. 57+6/
4. 57+6/41-2*
5. 57+6/41-2*^

[3+3 marks]

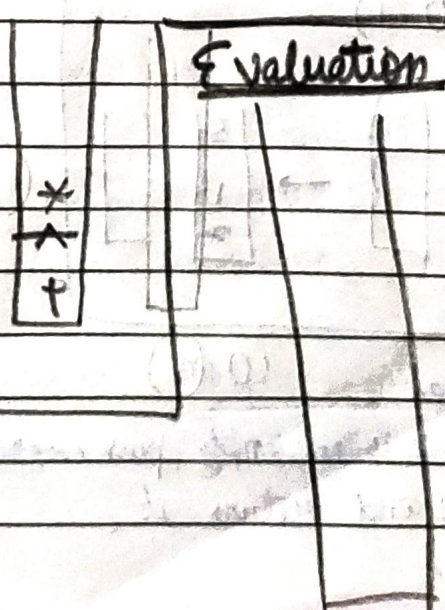
a) Convert following infix expression into Postfix expression using stacks.
Evaluate the resulting postfix expression too using stack.

5*6/2-4^3+8*5+7^2*2



5
56*
56*2/4
56*2/43^-
56*2/43^-8*
56*2/43^-85*72^2*+

Evaluation



'b)Implement a Queue using two stacks.

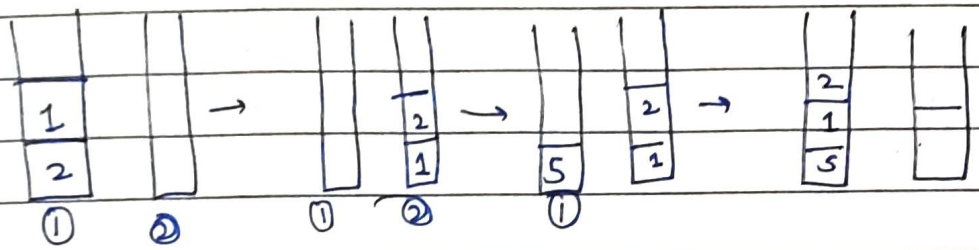
b) How can you implement a queue using two stacks? Explain with suitable examples. [3 marks]

7 7a] How do you model an queue using stack? Illustrate all possible steps.

## Method 1: By making enQueue costly

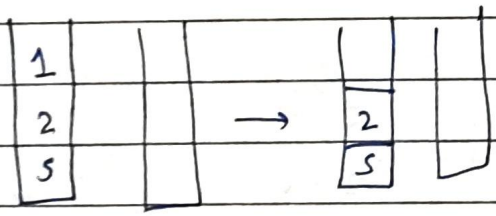
i) enqueue (q, x) O(n)

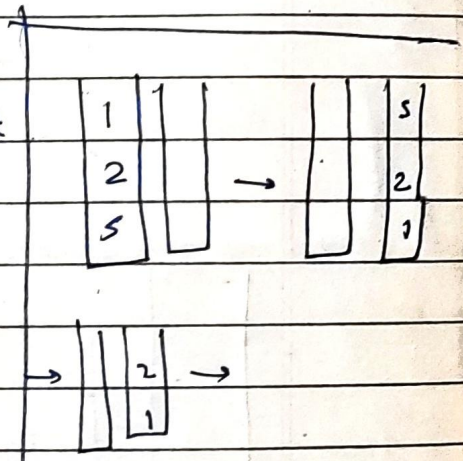
1. While Stack 1 is not empty, push everything from stack 1 to stack 2
2. Push x to stack 1
3. Push everything Back to Stack 1.




ii.) dequeue (q) O(1)

1. If Stack 1 is empty then error.
2. Pop an item from stack 1 and return it

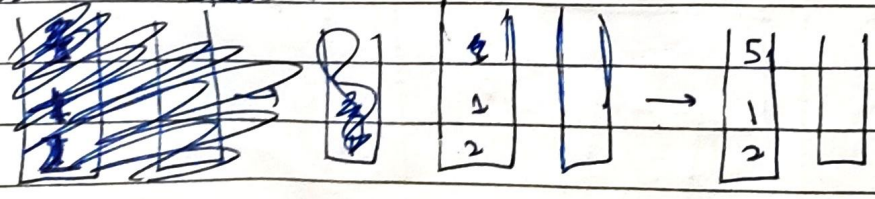



## Method 2: By making deQueue costly

i) enQueue (q, x) (O(1)

1. Push x to Stack 1




ii.) DeQueue (q)

1. If both Stack are empty then error (O(n)
2. If Stack 2 is empty: While Stack 2 is not empty, push everything from S1 → S2
3. Pop the element from Stack 2 and return it.

Date : / /
Page No.

CIRCULAR QUEUE

```
Class Queue {
  private:
  int total;
  int *arr;
  int size;
  int front;
  public:
  Queue (int cap){
      total = cap
      arr = new int [total];
      size = 0;
      front = 0;
  }
  void enqueue (int item)
  { if (isFull()) {
      cout << "Queue is full";
      return
    }
    int idx = (front + size) % total;
    arr [idx] = item;
    size++;
  }
  int dequeue ()
  { if (isEmpty ()) { cout << "Empty"; return; }
    int temp = arr [front];
    arr [front] = 0;
    front = (front + 1) % total;
    size--;
    return temp;
  }
  int getFront ()
  { if (isEmpty()) ——
    int temp = arr [front];
    return temp;
  }
  bool isEmpty ()
  { return size == 0; }
  bool isFull ()
  { return size == total; }
  void display (){
  for (int i=0; i < size; i++)
  { int idx = (i + front) % total;
    cout << arr [idx] << endl;
  }
  }
}
```

PSEUDO CODE

(ii) Insertion operation in a CIRCULAR QUEUE

B) In a circular queue, show the conditions that differentiate between empty queue and full queue. [2 marks]

...marks]

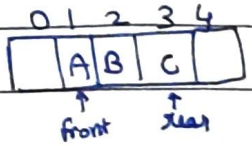
) Queue is set up in a circular array A[0...n - 1] with front and rear defined as usual. Assume that n – 1 locations in the array are available for storing the elements (with the other element being used to detect full/empty condition). Compute a formula to find the number of elements in the queue in terms of rear, front, and n.

Date : / /
Page No.

As per the question, the condition to detect empty or full queue is checked before using the formula.

∴ no. of elements = (n - front + rear + 1) % n

Eg.




Size : n = 5 | front = 1 | rear = 3 |

∴ # elements = (5 - 1 + 3 + 1) % 5 = 8 % 5 = 3

4. Modify the circular array-based queue implementation where one array position remains empty to distinguish between queue FULL and queue EMPTY conditions. Use a separate Boolean/Integer member to keep track of whether the queue is empty, rather than require that one array position remain empty. Use following Structure and write InsertQ() and DeleteQ() operations.

```
#define size 20;
struct Queue
{
    Int A[size];
    Int front, rear, IsEmpty;
};
```

(6 Marks)

no of elements % n = 0 → Full

→ Empty

[b] Explain the disadvantage of implementing linear queue using Array. Also show the conditions that differentiate Empty and Full Queue.

2

Date : / /
Page No.

On deletion of an element from existing Queue, front pointer is shifted to next position. This results into virtual deletion of an element. By doing this, the memory which was occupied by deleted element is wasted and hence insufficient memory utilization will occur.

Empty: front > rear OR front = -1

Full: rear = front - 1 OR rear = (size - 1) and front = 0

(b) Why circular queue is preferred over linear queue? Write down the underflow, overflow condition of Circular queue? Write down the algorithm for traversal of circular queue considering all cases. [1+1+1+1]

Full : $(n - front + rear + 1) \% n == n$ → TRUE

Empty : $(n - front + rear + 1) \% n == 0$ → TRUE

(e) For a circular queue, write the condition for "QUEUE_FULL" and "QUEUE_EMPTY".

Traversal :

```
for (int i=0 ; i<size ; i++)
{ int idx = (i + front) % total ;
  cout << arr[idx] << endl ;
}
```

Date : / /
Page No.

1) Write a program to delete redundant elements from a queue.

We will use sets otherwise code complexity ↑

```
void removeRedundant (queue <int> &q){
    set <int> s;
    int n = q.size();
    for (int i=0; i<n; ++i){
        int temp = q.front();
        q.pop;
        if (s.find(temp)!= s.end()) {continue;}
        else { s.insert (temp);
               q.push (temp);
        }
    }
}
```

Consider the following queue of characters where QUEUE is a circular array which is allocated six memory cells: (3 marks)

Front=2 Rear=4 QUEUE: _, A, C, D, _, _ ("_" denotes empty memory cells).

Describe the queue after each step as the following operations takes place:

i) F is added to the queue, ii) Two letters are deleted, iii) K, L, and M are added to queue, iv) Two letters are deleted, v) R is added to queue, vi) Two letters are deleted, vii) S is added to queue, viii) Two letters are deleted.

Date : / /

Page No.

| | FRONT | REAR | QUEUE |
|---|---|---|---|
| i) | 2 | 5 | _ _ A C D F _ |
| ii) | 4 | 5 | _ _ _ _ D F _ |
| iii) | 4 | 2 | L M _ D F K |
| iv) | 6 | 2 | L M _ _ _ K |
| v) | 6 | 3 | _ L M R _ _ K |
| vi) | 2 | 3 | _ M R _ _ _ |
| vii) | 2 | 4 | _ M R S _ _ |
| viii) | 4 | 4 | _ _ _ S _ _ |

c) What is the output of following function for start pointing to first node of following linked list? 1->2->3->4->5->6

```
void fun(struct node*start)
{
 if(start == NULL)
  return;
 printf("%d ", start->data);
 if(start->next != NULL )
  fun(start->next->next);
 printf("%d ", start->data);
}
```

OUTPUT

1 3 5 5 3 1

Date : / /
Page No.

Write the real world applications of STACK & QUEUE data structure.

| QUEUE | STACKS |
|---|---|
| → Q of people at a ticket window | → Reverse a word |
| → Vehicles on toll tax | → UNDO mechanism in OS |
| → Phone answering system | → Language processing. |

(b) The following C function takes a singly linked list of integers as a parameter and rearranges the elements of the list. The list is represented as pointer to structure. The function is called with the list containing integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes?

LIST → 2 1 4 3 6 5 7

```
struct node{
        int value;
        struct node *next;
        };
  void rearrange(struct node *list)
   {
      struct node *p, *q; int
      temp;
      if (list==NULL || list → next==NULL) return;
      p = list;
      q = list → next;
      while(q!=NULL)
     { temp = p → value;
      p → value = q → value;
      q → value = temp;
      p = q → next;
      if (p!=NULL) q=p → next;
         else q= NULL;
      }
   }
```

here consecutive elements pairs are getting swapped until q = NULL

Write an algorithm for returning value stored in $k^{th}$ node (from end) of the singly linked list.

→ maintain two pointers slow and fast.
→ move fast to the $k^{th}$ node from start
→ now move both slow and fast until fast reaches last node.
→ now print the data of the slow, as its at $k^{th}$ from end

CODE:

```
Kth from end (int k) {
    Node s = head;
    Node f = head;
    for (int i=0; i<k; i++) { f.next}
    while (f!= tail) {s=s.next; f=f.next; }
    return s.data;
}
```

1. Write a RemoveDuplicates() function which takes a singly linked list sorted in increasing order and deletes any duplicate nodes from the list. Ideally, the list should only be traversed once using fixed amount of extra storage. (5 Marks)

Date : / /
Page No.

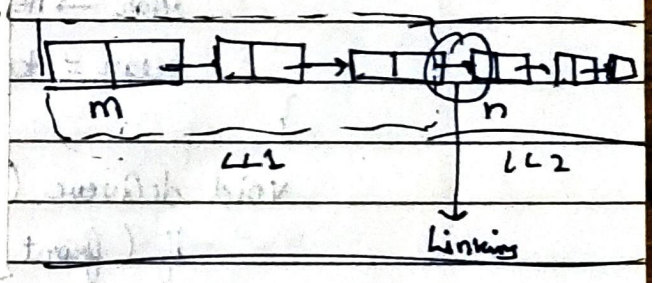
```
void removeDuplicates (Node *head)
{ Node * current = head;
  Node * next = next;
  if (current == NULL) { return ;}
  while (current → next != NULL) {
     if (current → data == current → next → data ) {
          next = next = current → next → next ;
          free (current → next);
          current → next = next = next;
     }
     else { current = current → next;}
     }
}
```

(d) What does the following function do for the given Linked Lists:

```
typedef struct node
{
   int data;
   node* next;
} node;
void join (node* m, node* n)
{
   node* p = n;
   while (p-> next ! = NULL)
   {
        p = p -> next;
   }
   p -> next = m;
}
```

Write an algorithm to concatenate two singly linked list?

The function two linked lists having heads at m and n.




a) Consider a linked list which has a loop in it. Write a pseudocode to check whether this linked list is either NULL-terminated or ends in a cycle (cyclic).

using Floyd's algorithm

```
bool detectCycle (node * &head){
    node * slow = head;
    node * fast = head;
    while (fast != NULL && fast → next != NULL) {
        slow = slow → next;
        fast = fast → next → next;
        if (fast == slow) { return true; }
    }
    return false;
```

Date : / /
Page No.

b) Give the linked implementation of a queue and discuss the implementation of priority queues. (3)

b) Implement a queue using a double linked list

```
Struct Node {
    int data ;
    Node * next ;
    Node (int d)
    { data =d ;
      next = NULL ;
    }
};
Struct Queue { Node * front , rear ;
    Queue () {
        front = rear = NULL ;
    }
    void enQueue (int x) {
        Node * temp = new Node (x) ;
        if (rear == NULL) { front = rear = temp ; return }
        rear → next = temp
        rear = temp ;
    }
    void deQueue () {
        if ( front == NULL) { return ; }
        Node * temp = front ;
        front = front → next ;
        if ( front == NULL)
            rear = NULL ;
        delete (temp) ;
    }
```

Date : / /

e No.

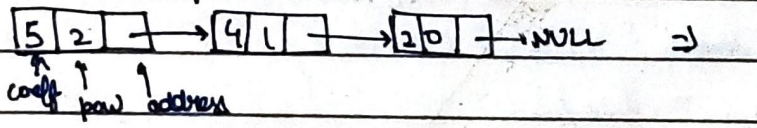
a) Write a program to insert a node with info 5 after each node having the info as 4 in a linked list. (4)

```
function (Node*head){

Node* temp = head;

for (int i=0; temp->next != NULL; i++){

If (temp->data == 4){

    Node* t = temp->next; Node* n = new Node(5);
    temp->next = n; n->next = t; }

}

}
```

[2] → [4] → [5]

a) Write a C program to add two polynomials using Linked lists (3)

Structure of link list

Eg) 

$\Rightarrow 5x^2 + 4x^1 + 2x^0$

CODE:

```
void add (struct Node * poly1, struct Node * poly2, struct Node * poly)
{ while (poly1->next && poly2->next) {
    if (poly1->pow > poly2->pow) { poly->pow = poly1->pow;
                                   poly->coeff = poly1->coeff;
                                   poly1 = poly1->next;
    elseif (poly1->pow < poly2->pow) { poly->pow = poly2->pow
                                   ——————— 2 ———————;
                                   — 2 ——— 2 ——— }
```

P.T.O.

Date : / /
Page No.

```
else { poly -> pow = poly1 -> pow ;
       poly -> coeff = poly1 -> coeff + poly2 -> coeff ;
       poly1 = poly1 -> next ;
       --- 2 ---- 2 ---- ;
}
{ poly -> next = (struct Node *) malloc (sizeof (struct Node)) ;
  poly = poly -> next ;                    // new node
  poly -> next = NULL ;
}
while (poly1 -> next || poly2 -> next) {
    if (poly1 -> next) { poly -> pow = poly1 -> pow ;
                         --- coeff ---- coeff --
                         poly1 = poly1 -> next ;
    }
    if (poly2 -> next){ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    }
    ~~poly -> next -> struct~~
    {
                                // new node
    }
}
}
```

What is meant by stack overflow condition? Is it applicable to the linked list method of implementation of the stack? Give reasons.

Stack overflow condition means that the no. of elements in the stack is equal to the size of array used to implement stack and no more elements can be pushed.

It is not applicable on linked list method since we can allocate new memory to every new element and thus no upper bound exist.

Date : / /

No.

2. Write an algorithm to merge two sorted linked lists into single sorted linked list without creating any new node.

```
Node * solve (Node * first, Node * second) {
    // only 1 ele in 1st list
    if (first -> next == NULL) { first -> next = second ; return first ; }
    Node * curr1 = first ;
    Node * next1 = curr1 -> next ;
    Node * curr2 = second ;
    Node * next2 = curr2 -> next ;
    while (next1 != NULL && curr2 != NULL) {
        if ((curr2 -> data >= curr1 -> data )
            && (curr2 -> data <= next1 -> data )) {
            curr1 -> next = curr2 ;
            next2 = curr2 -> next ;
            curr2 -> next = next2 ;
            curr1 = curr2
            curr2 = next2
        }
        else
        { curr1 = next1 ;
          next1 = next1 -> next ;
          if (next1 == NULL) { curr1 -> next = curr2 ; return first ; }
        }
    }
    return first ;
}
Node * sort ( Node * first, Node * second {
    if ( first == NULL) return second ;
    ~~second first ;~~
    if ( first -> data <= second -> data ) {
        return solve (first, second) ;
    } else {
        return solve (second, first) ;
    }
}
```

Date : / /

Page No.

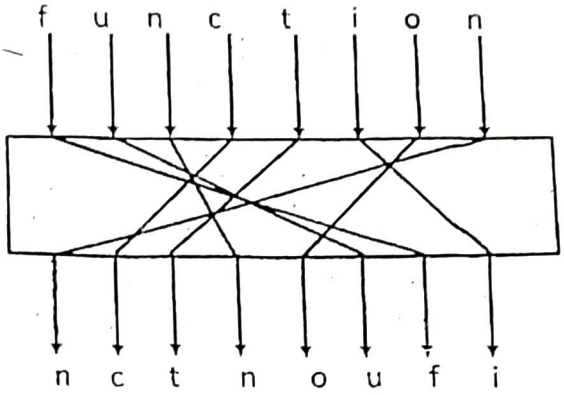
2. (a) Assume a set is represented by a linked list. Write down a best algorithm to perform union and intersection operation of two sets each represented by linked list containing n elements and find out the time complexity of each operation. [2+2+2=6]

```
Node* getUnion (Node* head1, Node* head2)
{  Node * ans = new Node (-1);
   Node * head = ans;
   Set <int> st;
   while (head1 != NULL){
            st. insert (head1 → data);
            head1 = head1 → next;
   }
   while (head2 != NULL) {
            st. insert (head2 → data);
            head2 = head2 → next;
   }
   for (auto it : st) { struct Node * t = new Node (it);
                        ans → next = t;
                        ans = ans → next;
   } head = head → next;   // to avoid -1
   return head
}

Node * getIntersection (  ,  ){
   Node * result = NULL;
   Node * t1 = head1;
   while (t1 != NULL) {
            if (isPresent (head2, t → data))
                     push (&result, t → data);
            t1 = t1 → next;
   }
   return resp.
}
```

TC  getUnion → O(m+n)

getIntersection → O(max)

Consider following diagram to rearrange characters of 8-char array (e.g. string




"function" is converted to string "nctnoufi".

(a) Design a suitable data structure (say 'X') and write function to perform this conversion.

(b) Write a function to find inverse of data structure 'X' such that string "nctnoufi" is converted back to "function" using this new structure.

Date : / /

Page No.

Date : / /
Page No.

) Write an algorithm to insert a node in the middle of a Doubly Linked List.

2. There is a singly linked list containing data sorted in non-decreasing order. There are two elements in this linked list which store similar data values. Write an algorithm to delete first node (out of two-containing equal values) which contains duplicate data value.

b) Write a program that creates a new single linked list by selecting alternate elements of a given linked list. (4)

What are priority queues. Discuss their implementation with r... to linked lists.

b) Consider an array of integers that contains both positive and negative numbers. Write a program to find two elements whose sum is closest to 0.

[3+3 m...

a) Describe the use of header nodes in linked lists and two different ways in which they can be defined (3)

(c) Define the following terms in tree data structure:
- Depth of a node
- Height of a node

END

(2 marks)

Let A and B be two linked list. Write a 'C' function to create a new linked list C that contains elements alternately from A and B beginning with the first element of A. If you run out of elements in one of the lists, the append the remaining elements of the other list to C (5 marks)

(3 marks)

...pointer 'q' consider all the cases. (2 marks)

Q4. A) Write an algorithm to form a doubly linked list from a one-way linked list. [3 marks]

B) Write an algorithm for deletion of an element from a linked list implementation of priority queue. [3 marks]

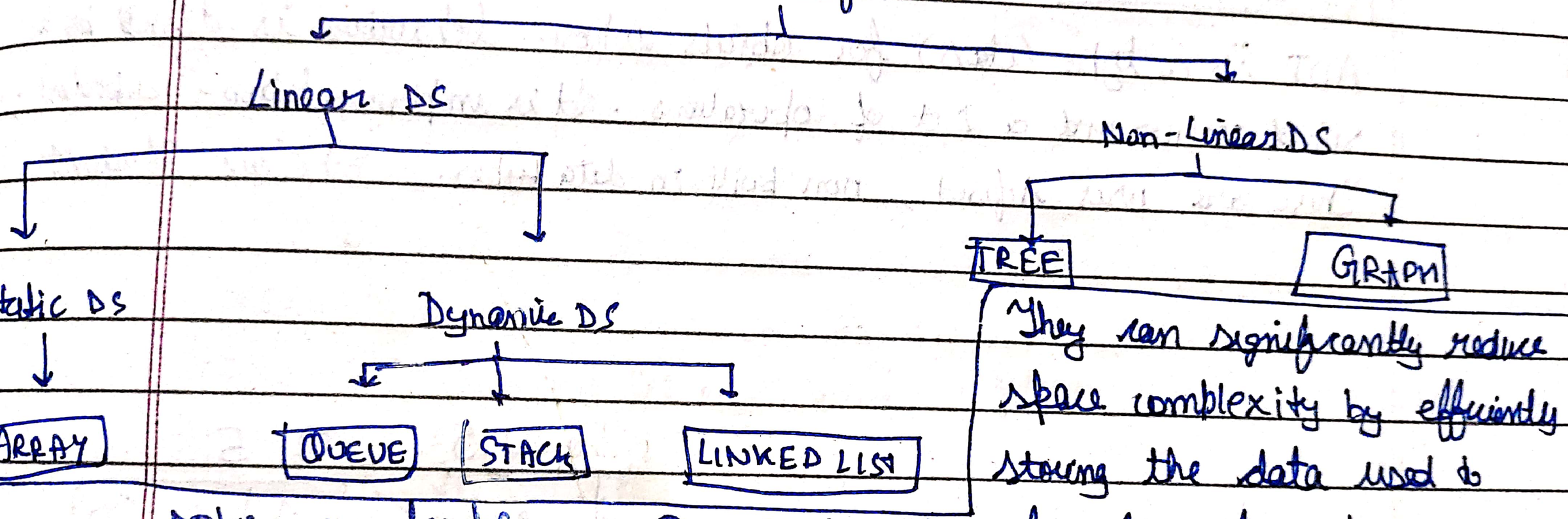
Q5. a) Describe the behaviour of quick sort algorithm when the input is already sorted. How this would be changed if instead of first element, we select the mid-point as the pivot point? [3 marks]

b) Sort the following list of elements using Heap Sort

12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18. [3 marks]

a) Classify different types of Data structures. Also discuss how they can help in reducing space complexity for different real world problems. (3)

Data structures are classified as.




They can significantly reduce space complexity by efficiently storing the data used to solve a problem. Eg:) Storing the friendlist of a person in the form of Graph (DS) is more efficient than creating a simple list.

b) Give the sequence of argument values that result when the following program is invoked for each of integers 3, 4 and 5. (4)

```
int puzzle(int n)
{
    If (n==1)
        return 1;
    If (n%2==0)
        return puzzle(n/2);
    else
        return puzzle(3*n+1);
}
```

| | |
|---|---|
| n=3 | 3 10 5 16 8 4 2 1 |
| n=4 | 4 2 1 |
| n=5 | 5 16 8 4 2 1 |

2)

a) What is the output of the following code snippet?

```
#include <stdio.h>
int main()
{
        int arr[][3]={{1,2},{3,4,5},{5}};
        printf("%d%d%d", sizeof(arr), arr[0][2], arr[1][2]);
        return 0;
}
```

i) ii) iii)

OUTPUT : 36 0 5

array is

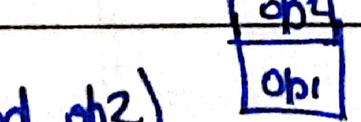
| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 2 | 0 |
| 1 | 3 | 4 | 5 |
| 2 | 5 | 0 | 0 |
| | 0 | 1 | 2 |

i) sizeof = 3×3×4 = 36

ii) iii) in diagram

Give an algorithm to convert a Prefix expression to an Infix expression using Stacks. (5)

1. Read the prefix expression in reverse order (right to left)
2. If symbol is an operand, push it into the stack.
3. If symbol is an operator (□), pop two operands (op1 and op2) and perform ans = op1 □ op2 (concatenate), and push ans in stack.
4. Repeat steps (2-3) until string is traversed
5. pop the last element of Stack as answer.



Date : / /

e No.

Write a program to insert a node with info 5 after each node having the info as 4 in a linked list. (4)

```
function ( Node* head ) {

Node* temp = head;

While(temp->next != NULL;) { }

If (temp->data == 4) {

    Node* t = temp->next; Node* n = new Node(5);
    temp->next = n ; n->next = t; }

}
```

12 → 4 → 8

Date : / /
Page No.

b) Implement a stack using arrays in C++

```
class Stack {
    private:
        int total;
        int *arr;
        int tos;
    public:
        Stack (int cap) {
            total = cap;
            arr = new int [total];
            tos = -1;
        }
        Void push (int item) {
            if (isFull()) {
                cout << "Full";
                return
            }
            tos++
            arr [tos] = item;
        }
```

(i) Deletion operation of STACKS

```
void pop() {
    if (isEmpty()) {
        cout<< "Empty";
    }
    int temp = arr [tos];
    arr [tos] = 0
    tos --
    return temp;
}
```

```
int peek() {
    if (isEmpty()) {
        return arr [tos];
    }
    int size() { return tos + 1; }
    bool isEmpty { return size() == total }
    void display() {
        for (int i = tos; i >= 0; i--) {
            cout << arr [i] << endl;
        }
    }
}
```

Write a program to delete redundant elements from a queue.

We will use sets otherwise code complexity ↑

```
void removeRedundent (queue <int> &q){
    Set <int> s;
    int n = q.size();
    for (int i=0; i<n; ++i){
        int temp = q.front();
        q.pop;
        if (s.find(temp)!= s.end()) { continue;}
        else { s.insert (temp);
               q.push (temp);
        }
    }
}
```

Page No.

b) Give the linked implementation of a queue and discuss the implementation of priority queues. (3)

b) Implement a queue using a double linked list

```
Struct Node {
    int data;
    Node * next;
    Node (int d)
    { data = d ;
      next = NULL ;
    }
};
Struct Queue { Node * front, *rear ;
    Queue () {
        front = rear = NULL;
    }
    void enQueue (int x) {
        Node * temp = new Node (x);
        if (rear == NULL) { front = rear = temp; return }
        rear → next = temp
        rear = temp ;
    }
    void deQueue () {
        if ( front == NULL) { return ; }
        Node * temp = front ;
        front = front → next ;
        if ( front == NULL)
            rear = NULL;
        delete (temp) ;
}
```

Total No. of Pages: 02

Roll No. .....

3rd Semester

(Sep-2022)

**B. Tech. [MC]**

**Mid Semester Examination**

***CS 251 Data Structure***

***Max. Marks: 20***

***Time 1 h 30 min.***

**NOTE:** *All questions are compulsory.*
*Assume suitable missing data if any.*

Q1 a) For the given function, answer the following questions: [3][CO4]

```
int Recursive(int first, int second)
{
 if(first == 1)
        return second;
 else
        printf("%d ", first + second);
        return Recursive(first - 2, first + second);
}
```

What will be the output of the above function for these function calls:

i) Recursive(11,12) ii) Recursive(4, 26)

b) Following function takes a Stack *S* as an argument, and uses a Queue *Q* to do processing. What does the below function do in general? [2][CO3]

```
void StackNQueue(Stack *S)
{
  Queue Q;
  while (!isEmpty(S))
  {
    InsertQueue(&Q, pop(&S));
  }
  while (!isEmpty(&Q))
  {
    Push(S, DeleteQueue(&Q));
  }
}
```

Q2. a) Convert following Infix expression into Postfix expression using stacks. Evaluate the resulting postfix expression using stacks too. [3][CO1]

9 * 6 + (8 / 4) – ((7+5) / 3) * 2

b) What data structure would you mostly likely see in a non-recursive implementation of a recursive algorithm? [1] [CO2]

P.T.O.

Q3. (A) The functionality of the following code is to reverse the elements of the linked list. State True or False with justification. [2][CO4]

```
void rearrange(struct node *list)
{
 struct node *p, * q;
 int temp;
 if ((!list) || !list->next)
   return;
 p = list;
 q = list->next;
 while(q)
 {
   temp = p->value;
   p->value = q->value;
   q->value = temp;
   p = q->next;
   q = p?p->next:0;
 }
}
```

b) Which of the following is false about a circular linked list? Justify. [2][CO3]

i. Every node has a successor.
ii. Time complexity of inserting a new node at the start of the list is O(1)
iii. Time complexity for deleting the last node is O(n).
iv. We can traverse the whole circular linked list by starting from any point.

c) What are the disadvantages of linked list in comparison to an array? [2] [CO1]

Q4. (A) Write an algorithm to delete an element from a linked representation of a priority queue. [2][CO3]

b) Write an algorithm to check if a singly linked list is palindrome. *You may use any one additional data structure, if required.* Palindrome is a sequence of characters which reads the same backward as forward. [3][CO3]

***All the Best***

Total No. of Pages: 2

Roll No. 2K15/MC/13

**III RD SEMESTER** **B.TECH. [MCE]**

**END SEMESTER EXAMINATION** **(October: - 2015)**

**CS-251 DATA STRUCTURES**

**Time: 3:00 Hours** **Max. Marks: 40**

**Note:** Answer any five questions
Assume suitable missing data, if any.

03-12-2016 20:13

1)

a) What is recursion? Explain with examples. (4)

b) Give the algorithm/pseudo-code to sort a list of numbers using quick sort. Show the steps of quick sort procedure for sorting the following list of numbers: 14, 26, 64, 56, 20, 3, 44 and 11. (4)

2)

a) Using examples discuss where the stack data structure will perform better than any other data structure like arrays and linked list. (4)

b) Write a routine/algorithm to find, count and remove duplicate elements in a queue. Give some applications of queues. (4)

3)

a) The order of the nodes of a Binary Tree in pre-order and inorder traversal are as follows:

**PREORDER:** A,B,L.M,K,N,P,Q

**INORDER:** L,B,M,A,N,K,Q,P

Draw the Tree. Also calculate its Post-order traversal. (4)

b) Write an Algorithm/routine to split a singly linked list from the middle. (4)

4)

a) Write a program in C/C++ to find out the maximum and minimum element in a binary search tree. (4)

b) What is a height balanced tree? Insert the following keys in a Binary search tree so that it is height balanced after each insertion.

44, 26, 13, 120, 99, 88, 64 (4)

5) a) Make a 4-way B-tree by inserting the following keys to an empty 4-way B-tree.

7, 15, 3, 16, 5, 1, 18, 10, 33, 28, 60, 17, 14 (4)

b) For the following key sequence determine the binary heap obtained when the keys are inserted one by one in the order given into an initially empty heap.

39, 19, 42, 91, 58, 70, 22, 16, 75, 40 (4)

6) a) Explain various techniques to represent the Graph in computer memory. Write Depth first Traversal (BFS) algorithm for traversing a graph and also explain it with an example. (4)

b) Draw a hash table with open addressing and a size of 9. Use the hash function "k%9". Insert the keys: 5, 29, 20, 0, 27 and 18 into your table. (4)

7) Write short notes :- (4x4)

a) Binary search tree

b) Threaded Binary Trees

c) Doubly Circular Linked List

d) Priority Queues using heaps

*Total No. of Pages 1* *Roll No.....*

IIIRD SEMESTER

B.Tech.(Computer Engg.)

MID SEMESTER EXAMINATION (Sept. – 2022)

Course Code: COE-201 Course title: Data Structures

Time: 1:30 Hours Max. Marks: 20

| Note: | Answer all questions. Use C codes /pseudo codes for you answers. Assume suitable missing data, if any. |
|---|---|

1. A set is stored as an unordered linked list. Write a method to compute union of two sets s1, and s2 represented as linked lists. Your method should not destroy linked lists s1 and s2. Also compute time complexity of you algorithm.

[4+1][CO2)

2. Write an algorithm for returning value stored in $k^{th}$ node (from end) of the singly linked list. (return first node value if linked list length is less than k and -1 if linked list is empty).

[4][CO2]

3. Consider strings constructed using only the following characters '(', ')', '[', and ']'. Character '(' matches with only ')' and '[' matches with only ']'. A string comprising of these characters is complete if every character is followed by its matching character or a substring that is complete followed by its matching character. For example "()", "[]", "([])", "[]([]())" are all examples of complete strings, while "([)]" and "[][(])" are examples of incomplete strings.
Write algorithm/code using character stack which accepts such strings as input and prints "Complete" or "Incomplete".

[4][CO1]

4. A queue is implemented using array. We need to modify behavior of queue. When a request to delete the front element X is called and X is being deleted first time, then element X is given a second chance. Here, X is removed from front side and inserted at rear side. But when X is the front element and a delete request comes, and it is second time a delete request comes and X is at front, it is removed from queue. Define modified queue data structure and write codes/pseudo codes for InsertQ() and DeleteQ() methods.

[4][CO1]

5. Consider a 3-dimensional array A[12][5][10] and base address of array is 20000. If this array is stored in row-major form, calculate the address of array element A[6][3][5]. Remember, array index starts from 0.

[3][CO1]

# MCE DSA (Repeated Question or Imp Ques)

Q2. A) Compute the postfix equivalent of the following infix arithmetic expression

$$((5+7)/6)\uparrow((4-1)*2)$$

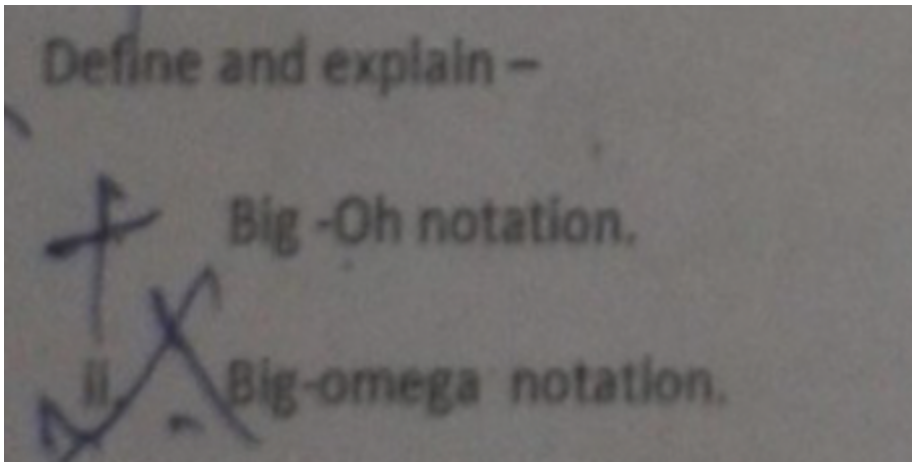
Where ↑ represents exponentiation. Assume normal operator precedence. Also evaluate the generated postfix expression using stack. Show status of stack after each execution. [4 marks]

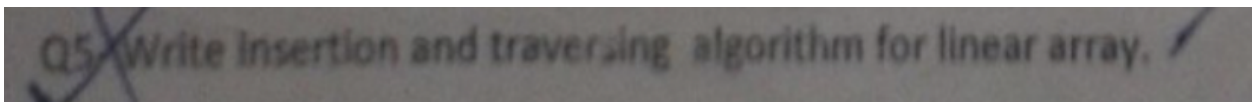
B) In a circular queue, show the conditions that differentiate between empty queue and full queue. [2 marks]

[3 marks]

Q4. A) Write an algorithm to form a doubly linked list from a one-way linked list. [3 marks]

B) Write an algorithm for deletion of an element from a linked list implementation of priority queue. [3 marks]



1[a] How do you measure the complexity of an Algorithm Find out the complexity for the followings:

(i) Linear search (Best, Average & Worst case)

(ii) Binary search (Worst case)

(iii) Bubble sort (Worst case) 3

Q6- Using insertion sort find out the value after 5th iteration of following array

40,50,12,30,90,18,06,60,

A 2D Array DATA [6][8] is stored in a column major order with base address 301 and size 1 calculate the address of DATA [2][4]. 2

Write a recursive function to count the number of occurrences of the number 24 in an array of n integers. 2

OR

(a) Consider a linked list which has a loop in it. Write a pseudocode to check whether this linked list is either NULL-terminated or ends in a cycle (cyclic).

(b) Write an algorithm to insert a node in the middle of a Doubly Linked List.

*Total No. of Pages 2*
**IIIRD SEMESTER**
**SUPPL. EXAMINATION**

*Roll No.....*
**B.Tech.(Computer Engg.)**
**(Feb.–2019)**

Paper Code: **COE-201**
Title of the subject: **Data Structures**
**Time: 3:00 Hours**
**Max. Marks: 40**

**Note:** Answer any five questions. Write pseudo code/C code for all algorithms asked. Assume suitable missing data, if any.

1. (a) Write an algorithm to evaluate a postfix expression.

   (b) Consider two strings $X=x_1,x_2,..x_m$ and $Y=y_1, y_2,...y_n$ where $x_i$, $1<=i<=m$ and $y_i$, $1<=i<=n$ are members of finite set symbols. Write an algorithm to generate a string by taking 1 element from each list. When any one string is exhausted, the output string should store rest of the elements of other string. **[4,4]**

2. Consider a list of numbers: 62, 31, 70, 91, 25, 11, 9, 61, 73, 6
   Write an algorithm to convert this array into a Max-Heap and show the application of the algorithm on given array/list. Show heap construction after every swap operation. **[8]**

3. Let the key of a node in a binary search tree be X (let's also call this node, "node X"). Please give a definition of inorder Predecessor(X), and inorder Successor(X). Given that you are at node X write algorithm Predecessor(X), and Successor(X). Assume each node is having a parent pointer and root node address is always available. **[8]**

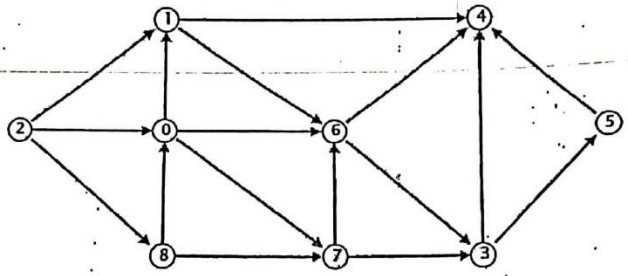
4. (a) Given two linked lists a and b, each containing n distinct numbers, design two different algorithms (possibly with different efficiency) to determine whether the two lists contains precisely the same set of numbers (but possibly in a different order).

   (b) Write an algorithm to reverse a singly linked list. **[6,2]**

5. A priority queue is a data structure that supports storing a set of values, each of which has an associated key. Each key-value pair is an entry in the priority queue. The basic operations on a priority queue are: insert(k, v): insert value v with key k into the priority queue, removeMin(): return and remove from the priority queue the entry with the smallest key. Write complete implementation of this priority queue. **[8]**

P.T.O

6. Consider the following acyclic digraph. Assume the adjacency lists are in sorted order: for example, when iterating through the edges pointing from 0, consider the edge $0 \rightarrow 1$ before $0 \rightarrow 6$ or $0 \rightarrow 7$.




Give topological sorting order for this graph. Also give DFS and BFS output starting from vertex 2. **[4+2+2]**

7. (a) Explain properties and structure of a B-tree. Draw a B-tree of degree 4 or more having atleast three levels.

(b) Explain BFS graph traversal technique. Write an algorithm for BFS traversal such that along with traversal it also computes single source shortest path for a given unweighted graph. **[4,4]**

8. (a) Write an algorithm to count number of non-leaf nodes (internal nodes) in a given binary tree.

(b) Write an algorithm to add two polynomials using array of structures. **[4,4]**

Total No. of Pages 2

Roll No..... .....

IIIRD SEMESTER

B.Tech. (Computer Engg.)

END SEM. EXAMINATION

(Nov-Dec. – 2019)

Paper Code: CO201

Title of the subject: **Data Structures**

Time: 3:00 Hours

Max. Marks: 40

**Note:** Answer any five questions. Write pseudo code/C code for all algorithms asked. Assume suitable missing data, if any. Write your assumption(s) in answer.

1. (a) Given an array A, which was obtained by swapping two elements (selected randomly) of a sorted array e.g. {5, 45, 17, 20, 32, 12, 67}. Write an algorithm to get back the original sorted array by just swapping one pair.

   (b) Write a 3-way merge program/algorithm to merge three sorted arrays into a single sorted array. [4,4]

2. (a) Implement a function [int successor(TreeNodeptr x)] that returns the successor of a node in a binary search tree (the BST stores integer keys). A successor of a node n is defined as the smallest key x in the BST such that x is bigger than the value of n, or null if that does not exist. You may assume that the BST does not contain duplicate keys. Note that getLeft(), getRight(), and getParent() return null if the node does not have a left, a right child, or is the root, respectively.

   (b) Develop an efficient algorithm called Partition-Even-Odd(A,n) that partitions an array A[0..n-1] in even and odd numbers. The algorithm must terminate with A containing all its even elements preceding all its odd elements. Partition-Even-Odd may use only a constant memory space in addition to A. In practice, this means that you may not use another temporary array. [4+4]

3. (a) Consider two shorted lists of length m and n respectively. A merge algorithm is applied to merge these lists into a single sorted array of size m+n (assuming all number are unique). Give exact number of comparisons in best case and in worst case.

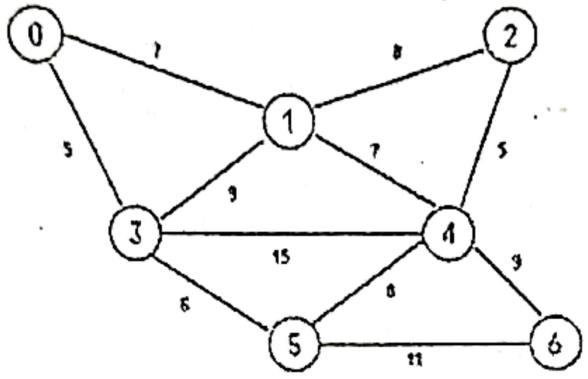
   (b) Write an algorithm (preferably recursive and linear time) to delete all nodes of a linked list and set the pointer (initially pointing to first node) to NULL. Remember we need to physically delete each node and free the memory occupied by each node. [4,4]

4. (a) Write an algorithm to check if given binary tree is a Binary Search Tree or not.

(b) Suppose that we insert a new data x into a Binary Search tree and then immediately delete x from the tree. Will the new tree be identical to the original one? If yes give the reason in no more than 3 sentences. If no give a counterexample. Draw pictures if you necessary.

[4,4]

5. Write Prim's Algorithm for MST. Apply Prim's MST algorithm on following graph and find Minimum Spanning Tree. Show each step of the algorithm.

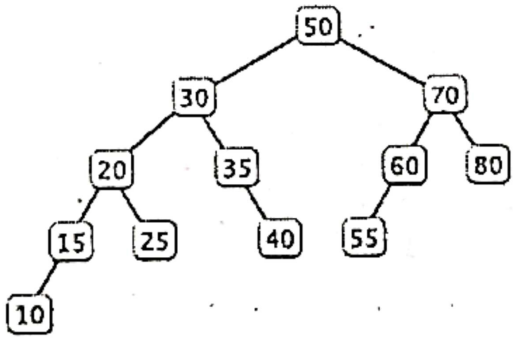



[8]

6. (a) Write algorithm to print all connected components of a given undirected graph.

(b) Write an algorithm to find kth element/node from end in a singly linked list using two pointers and one pass/scan of the linked list). Assume k>=1 and linked list is having more than k nodes. [4,4]

7. (a). Consider following height balanced tree (AVL tree)




Draw the tree after insertion of each of the following data in given order: 5, 45, 53.

(b) Explain BFS graph traversal technique. Write an algorithm for BFS traversal such that along with traversal it also computes single source shortest path for a given unweighted graph. Apply this algorithm on graph given in Question 5 ignoring weights and starting from vertex '0'. [4,4]

Total No. of Pages: 02

Roll No. ... .....

B. Tech. [MC]

3rd Semester

Mid Semester Examination

(September-2018)

CS251 Data Structure

Time 1h 30 min.

Max. Marks: 30

NOTE: *Attempt all Questions. Assume suitable missing data if any.*

Q1 A) An m*n matrix is said to have a saddle point if some entry a[i][j] is the smallest value in row I and largest value in column j. Write an algorithm that determines the location of a saddle point if one exists. [3 marks]

B) Explain average case time complexity of linear search. [3 marks]

Q2. A) Compute the postfix equivalent of the following infix arithmetic expression

$$((5+7)/6)\uparrow((4-1)*2)$$

Where ↑ represents exponentiation. Assume normal operator precedence. Also evaluate the generated postfix expression using stack. Show status of stack after each execution. [4 marks]

B) In a circular queue, show the conditions that differentiate between empty queue and full queue. [2 marks]

Q3. A) Suppose the following stack of integers is in memory where STACK is allocated N=6 memory cells: [3 marks]

TOP =2 STACK: 5, 2, 3, ___, ___, ___

Find the output of following program segment:

1. Call POP(STACK, ITEMA)

   Call POP(STACK, ITEMB)

   Call PUSH(STACK, ITEMB + 2)

P.T.O

Call PUSH(STACK, 8)

Call PUSH(STACK, ITEMA + ITEMB)

2. Repeat while TOP ≠ -1:

Call POP(STACK, ITEM)

Write: ITEM.

[End of loop]

b) How can you implement a queue using two stacks? Explain with suitable examples.

[3 marks]

Q4. A) Write an algorithm to form a doubly linked list from a one-way linked list.

[3 marks]

B) Write an algorithm for deletion of an element from a linked list implementation of priority queue.

[3 marks]

Q5. a) Describe the behaviour of quick sort algorithm when the input is already sorted. How this would be changed if instead of first element, we select the mid-point as the pivot point?

[3 marks]

b) Sort the following list of elements using Heap Sort

12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18.

[3 marks]

**END**

*Total No. of Pages:3*

IIIrd SEMESTER

END SEMESTER EXAMINATION

*Roll No.....*

B.Tech.(COE/SE)

(NOV. – 2018)

Paper Code: CO-201

Time: 3:00 Hours

Title: Data Structures

Max. Marks: 40

**Note:** 1. Attempt all questions. Assume any suitable value(s) for missing data.
2. If asked to write algorithms, write as C functions or pseudo codes.
3. If any assumption is made for any question, write it.

1. (a) Consider an array A of integers. Suppose that A.length returns the length of such an array.

   (i) Write down an iterative algorithm in pseudocode that returns true if the array is sorted and false otherwise.

   (ii) Write a recursive algorithm following the divide-and-conquer approach (where problem size is reduced in each recursive call) that tests whether A is sorted. Start from the fragment below.

   (4+4=8)

2. (a) A connected component of a graph is a set of vertices where each node can reach every other node in the component along the given edges, and which is connected to no additional vertices. Give a modified version of Kruskal's algorithm to compute the number of connected components in a graph.

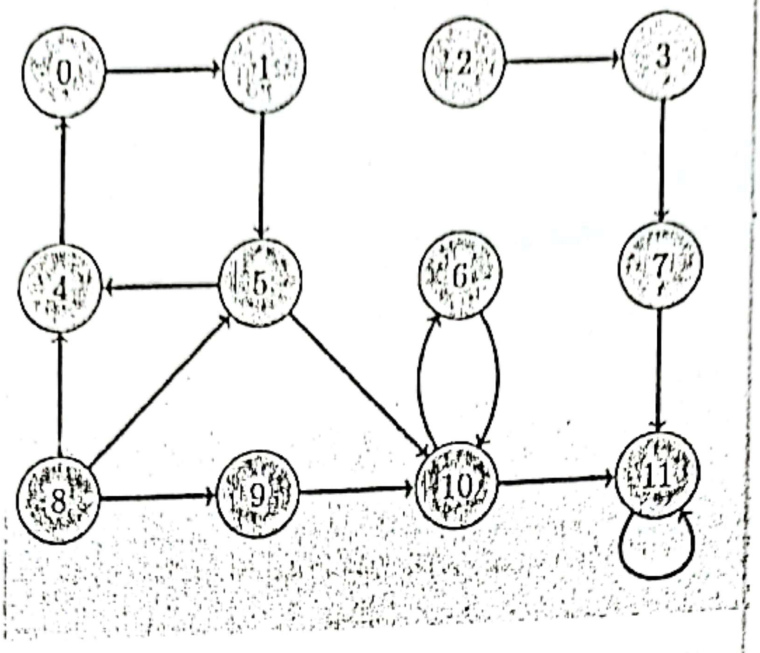
   (b) Suppose that we have numbers between 1 and 100 in a binary search tree and want to search for the number 45. Which (possibly multiple) of the following sequences could be the sequence of nodes examined?

   - 5, 2, 1, 10, 39, 34, 77, 63.
   - 1, 2, 3, 4, 5, 6, 7, 8.
   - 9, 8, 63, 0, 4, 3, 2, 1.
   - 8, 7, 6, 5, 4, 3, 2, 1.
   - 50, 25, 26, 27, 40, 44, 42.
   - 50, 25, 26, 27, 40, 44.

   (4+4=8)

3. (a) Suppose we have a MaxHeap H and two values $v_1$ and $v_2$, such that all values are distinct. Let $H_{12}$ be the heap you get if you insert $v_1$ and then $v_2$ into H, and $H_{21}$ be the heap you get if you insert $v_2$ and then $v_1$ into H. Give an example of H, $v_1$ and $v_2$ such that $H_{12} \neq H_{21}$. No justification needed, just draw the heaps H, $H_{12}$ and $H_{21}$.

   (b) Give one possible DFS traversal starting from node 0 of the graph below, printing the nodes both as they are discovered and finished. Also classify all edges into 4 classes (tree, Back, forward, cross). Note that the graph is directed.




(4+4=8)

4. (a) Since storing a triangular matrix as a two dimensional array wastes space, we would like to find a way to store only the nonzero terms in the triangular matrix. Find an addressing formula for the elements $L_{i,j}$ to be stored in a one dimensional array a. In other words, where in the array a would you store the element $L_{i,j}$? (You only need to explain how to map each nonzero element to the one dimensional array. No need to write down the actual function.)

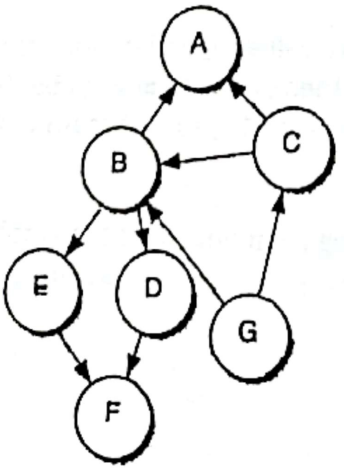
(b) Given an array of N elements of three different types: Cold, Warm, and Hot, design and describe clearly an O(N) in-place algorithm to put all the cold elements, on the left, followed by all the warm elements, followed by all the hot elements on the right. Your algorithm can use only a small **constant amount of extra space.** Show how your algorithm would operate on following array:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| C | W | H | C | W | W | C | H | C | W | C |

(4+4=8)

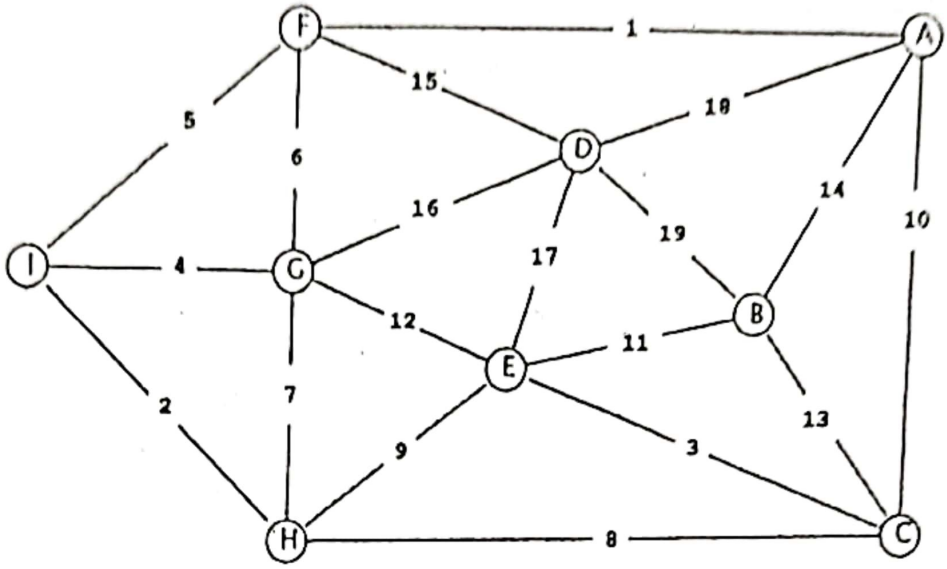
5. (a) Draw the B-trees that result when adding the following values in succession, starting with an empty tree. Assume each node can only hold 2 keys. To save drawing time, you can choose to only draw a new tree when a split occurs, but make it clear which value caused the split. Values: 35, 87, 64, 78, 81, 85, 22, 31.

(b) Find a topological ordering of the following graph?

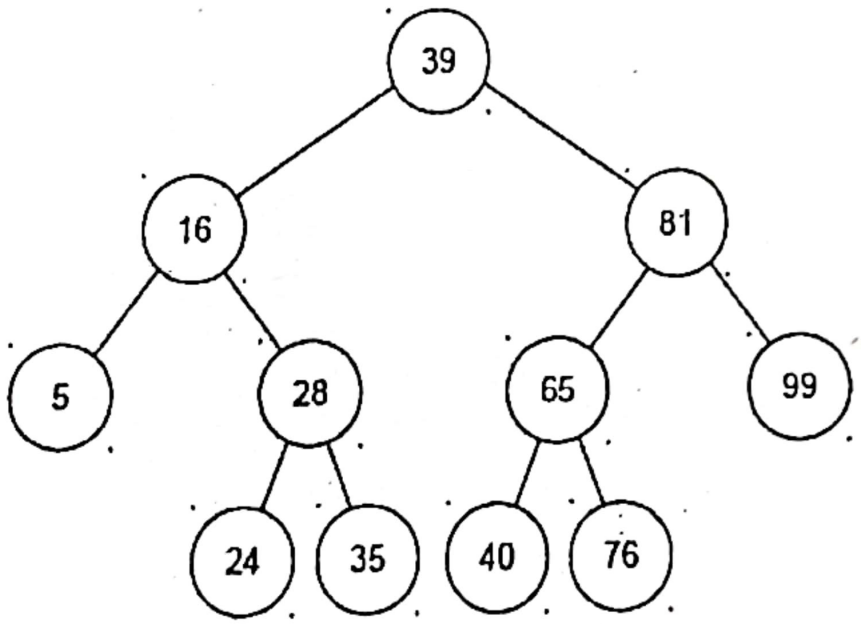



(4+4=8)

6. (a) Compute MST for the given graph using Prim's algorithm showing graph after each iteration of the algorithm.




(b) Consider following AVL tree. Draw AVL tree after inserting a new key 25.




(5+3=8)

7. (a) Given two arrays a[] and b[], each containing n distinct numbers, design two algorithms (with different performance) to determine whether the two arrays contains precisely the same set of numbers (but possibly in a different order).

(b) Write insertion sort algorithm for sorting a linked list into ascending order.

(4+4=8)

8. (a) Write algorithm count_LEAF() to count number of leaf nodes in a given binary tree.

(b) Write an algorithm isBST() to check if given binary tree is a Binary Search Tree.

(4+4=8)

Old

Total No. of Pages 3 Roll No.....

IIIRD SEMESTER B.Tech.(Computer Engg.)

END SEMESTER EXAMINATION (Nov. – 2018)

Paper Code: COE-203 Title of the subject: Data Structures

Time: 3:00 Hours Max. Marks: 70

**Note:** Answer any five questions. Write pseudo code/C code for all algorithms asked. Assume suitable missing data, if any.

1. (a) Implement insertion sort using linked list to perform insertion sort as follows: remove an element from the input list and insert it to the correct position in the output linked list. Write an algorithm for this insertion sort to sort n elements stored in a linked list.

   (b) Describe Overflow (stack full) and Underflow (stack empty) conditions in STACK data structure implemented (i) using array, (ii) using linked list. [8,3+3]

2. Consider a list of numbers: 62, 31, 70, 91, 25, 11, 9, 61, 73, 6
   Show the result of inserting the numbers in the list in the same order specified above into an initially empty **minimum heap**. Note that you need to show how the heap looks like after each number is inserted. [14]

3. (a) You are planning the seating arrangement for a wedding given a list of guests, V. Suppose you are also given a lookup table T where T[u] for u ∈ V is a list of guests that u knows. If u knows v, then v knows u. You are required to arrange the seating such that any guest at a table knows every other guest sitting at the same table either directly or through some other guests sitting at the same table. For example, if x knows y, and y knows z, then x, y, z can sit at the same table. Describe an efficient algorithm that, given V and T, returns the minimum number of tables needed to achieve this requirement. Analyze the running time of your algorithm.

   (b) Describe different methods for handling collision in Hash table. [10,4]

4. (a) Write an algorithm which checks if a given binary tree is a binary search tree (BST) or not. Algorithm should return true/false.

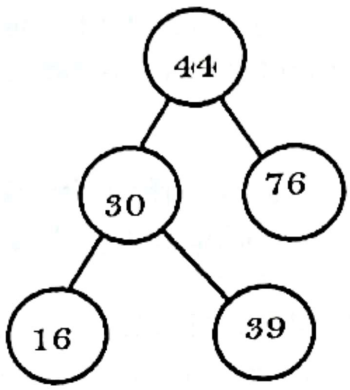
   (b) Write an algorithm that takes two linked lists, sorted in increasing order and merge the two into one sorted linked list, and return it.

   [7,7]

5. Write algorithm to evaluate a postfix expression using suitable data structure. Your algorithm should detect and display error messages if given postfix expression in not valid/correct. **[14]**

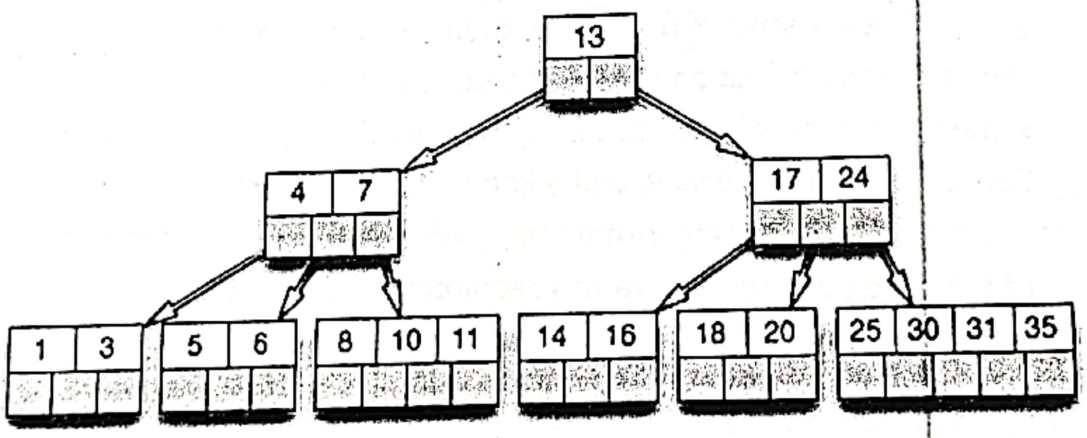
6. (a) Write algorithm which takes input a binary search tree T1 and returns a duplicate tree T2.

(b) Insert key 11 in following AVL tree and draw final AVL tree after required rotation(s) if any.




**[7,7]**

7. (a) Consider following B-Tree of degree 6 (max 6 children)




(i) Draw tree after deleting key=6.

(ii) Draw tree after deleting key =16 (without applying (i) i.e. key=6 is still present in tree).

(b) A given string contains all types of printable characters. Write an algorithm to remove all characters other than alphabets from string with fixed amount of additional storage area. (e.g. "He4l*l%=O" to be converted to "HellO"). What is time complexity of your algorithm? **[4+4, 6]**

8. (a) Write an algorithm to count number of leaf nodes in a given binary tree.

(b) Write an algorithm to concatenate two doubly circular linked lists L1 and L2. L1 and L2 are pointers to first node of linked lists respectively. After concatenation (L1 followed by L2), L1 points to first node of final linked list.

**[7,7]**

Old

*Total No. of Pages 1* *Roll No. .....*

3rd SEMESTER **B.Tech.**

SUPPLEMENTARY EXAMINATION ***Feb-2018***

**PAPER CODE: CO/SE-203** **TITLE OF PAPER: Data Structures**

*Time: 3:00 Hours* ***Max. Marks : 70***

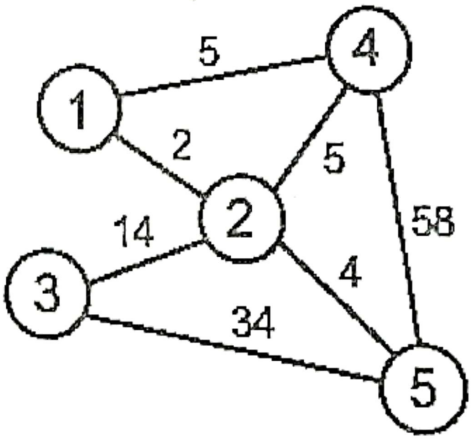
Note : Attempt any 5 questions.
All questions carry equal marks.
Assume suitable missing data, if any.

Que 1. (a) Write a program to check if a given Binary Tree is Heap.
(b) Define Tree, Binary Tree and Max-Heap.

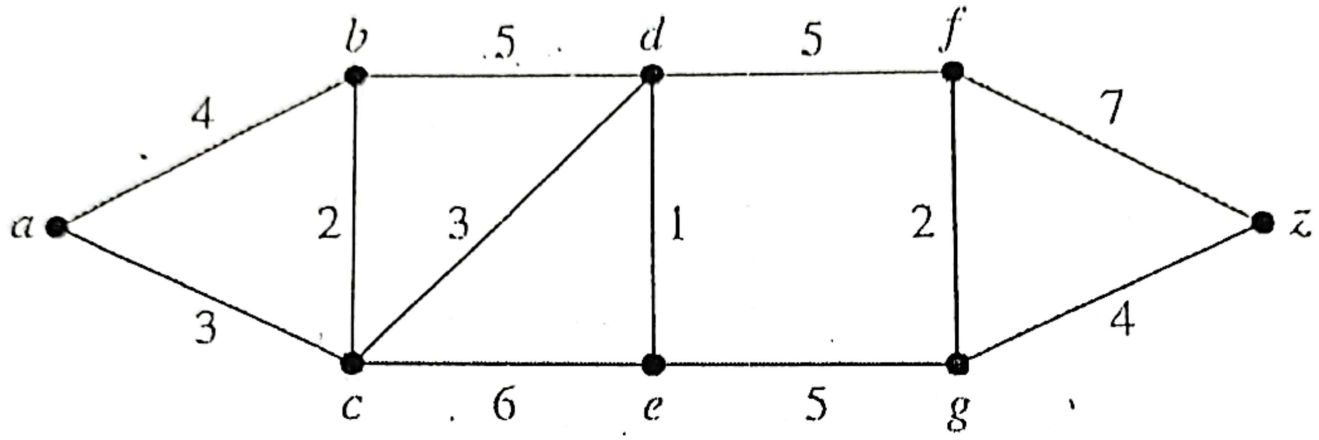
Que 2. (a) There are two linked lists A and B containing the following data:
A: 3,7,10,15,16,9,22,17,32
B: 16,2,9,13,37,8,10,1,28
Write a program in C to create a Linked list D which contains all elements of A as well as B ensuring that there is no repetition of elements.

(b) For a Strictly Binary Tree, Number of Leaf nodes= Number of Internal Nodes +1. Prove it.

Que 3. (a) Write an algorithm for insertion operation in BST. Give BST formed by inserting following elements in the given order: 20 17 6 8 10 7 18 13 12 5.
(b) Define Minimum Spanning Tree. Find MST for the following graph:



Q.5 [a] Apply Dijkstra's Algorithm to find shortest path from a to z in the network below: (5)




[b]

i. State and Prove the Handshaking Theorem.

ii. Determine the maximum number of vertices and edges in a simple graph with $n$ vertices. (3+2)

Q.6 [a] Prove that a graph is Euler if and only if every vertex is of even degree. (5)

[b] What is Chromatic Number. Find the chromatic number of : (5)

i) A complete graph of $n$ vertices, $K_n$.

ii) A bipartite graph

*Total No. of Pages: 02*

**III Semester**

**Mid Semester Examination**

Roll No. .....

**B. Tech. [MC]**

*(Sept-2019)*

*CS 251 Data Structure*

*Time 1 h 30 min.* *Max. Marks: 30*

**NOTE:** *All questions are compulsory.*
*Assume suitable missing data if any.*

Q1. [6 marks]

(a) Let A be a two-dimensional array declared as follows:

A: array[1...10] [1...15] of integer;

Assuming that each integer takes one memory location. The array is sorted in row-major order and the first element of the array is stored at location 100, what is the address of the element A[i][j] ?

(b) Convert the given Infix expression into its equivalent Prefix and Postfix expression:

((A+B)^C)*(D-E)^(F/G))

(c) A single array A[1...MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (top1<top2) point to the location of the topmost element in each of the stacks. What is the condition for "stack full"?

(d) What does the following function do for the given Linked Lists:

```
typedef struct node
{
   int data;
   node* next;
} node;
void join (node* m, node* n)
{
   node* p = n;
   while (p-> next ! = NULL)
   {
         p = p -> next;
   }
   p -> next = m;
}
```

(e) For a circular queue, write the condition for "QUEUE_FULL" and "QUEUE_EMPTY".

(f) What does the following fragment of C-program print:

```
char c[] = "IATE2019";
char *p = c;
printf("%s", p + p[3] - p[1]);
```

Q2. [3+3 marks]

(a) Suppose you are given an array s[1...n] and a procedure reverse(s,i,j) which reverses the order of elements in between positions i an j (both inclusive). What does the following sequence do, where $1 \leq k \leq n$:

```
reverse(s,1,k);
reverse(s,k+1,n);
reverse(s,1,n);
```

Explain with the help of an example.

(b) Consider an array of integers that contains both positive and negative numbers. Write a program to find two elements whose sum is closest to 0.

Q3. [3+3 marks]

(a) Queue is set up in a circular array A[0...n - 1] with front and rear defined as usual. Assume that n – 1 locations in the array are available for storing the elements (with the other element being used to detect full/empty condition). Compute a formula to find the number of elements in the queue in terms of rear, front, and n.

(b) Implement a Queue using two stacks.

Q4. [3+3 marks]

(a) Convert following Infix expression into Postfix expression using stacks. Evaluate the resulting postfix expression too using stack.

5*6/2-4^3+8*5+7^2*2

(b) What is Recursion? Which data structure is used to perform recursion? Explain with the help of an example.

Q5. [3+3 marks]

(a) Consider a linked list which has a loop in it. Write a pseudocode to check whether this linked list is either NULL-terminated or ends in a cycle (cyclic).

(b) Write an algorithm to insert a node in the middle of a Doubly Linked List.

Total No. of Pages 2 Roll No. MC-059.....

**III RD SEMESTER** **B.Tech.[MC]**

**END SEMESTER EXAMINATION** **(Nov-2014)**

***MC-204*** **DATA STRUCTURES**

**Time: 3:00 Hours** **Max. Marks : 70**

**Note:Answer any FIVE questions. All Questions carry Equal Marks.**

**Assume suitable missing data, if any.**

Q.1(a)Explain Asymptotic Notations for Complexity of Algorithms:
Omega(Ω), Theta(Θ) and Big O. (7)

(b) What is Reverse Polish Notation? Evaluate the following postfix expression using STACK:
12, 2, 5, ↑, +, 13, 2, -, /, 1, + (7)

Q.2(a) Explain Heap-sort Algorithm with the help of suitable example. (7)

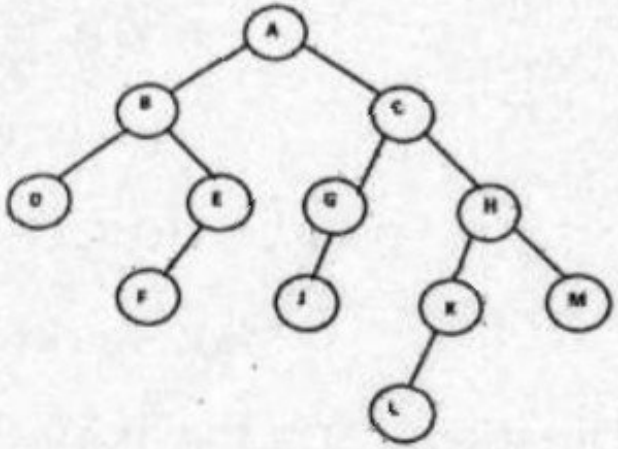
(b) Sort the following array using Merge Sort:
66, 33, 40, 22, 55, 88, 60, 11, 80, 20, 50, 44, 77, 3
Mentions the required steps to solve the question. (7)

Q.3(a) Compare Binary Tree, Complete Binary Tree , Extended Binary Tree, AVL trees and B-trees with the help of an example. (7)

(b) Write an Algorithm/Pseudocode to insert a node at Binary Search Tree with the help of suitable example. (7)

Q.4(a) Briefly explain the Linked and Sequential Representation of trees in detail followed by suitable examples and figures if needed. (7)

(b) Write a stepwise procedure to find Postorder traversals of the following tree using STACK:




(7)

Q.5(a) Write a difference between Multi-graph and Directed Acyclic Graph with the help suitable example. (7)

(b) Explain Prim's Algorithm with the help of suitable example. (7)

Q.6 Explain Warshall Algorithm to find a shortest path in detail with the help of a suitable example. (14)

Q.7 Write a short note on any TWO: (14)

(i) Memory Allocation and Garbage Collection
(ii) File Organization and Indexing
(iii) Directed Acyclic Graphs(DAG) and Topological Sort
(iv) DFS and BFS traversals of Graph

*Total No. of Pages 1*

*Roll No. .....*

**THIRD SEMESTER**

**B.Tech. (MC)**

**MID SEMESTER EXAMINATION** ***September-2012***

**MC-204 DATA STRUCTURE**

***Time: 1Hour 30 Minutes*** ***Max. Marks : 20***

**Note :** Answer *ALL* questions.
Assume suitable missing data, if any.

1[a] How do you measure the complexity of an Algorithm Find out the complexity for the followings:

(i) Linear search (Best, Average & Worst case)

(ii) Binary search (Worst case)

(iii) Bubble sort (Worst case) 3

[b] Explain the disadvantage of implementing linear queue using Array. Also show the conditions that differentiate Empty and Full Queue. 2

2[a] Write an algorithm to convert infix expression to reverse police notation using underlying Data Strcuture STACKS. Also illustrate the steps for the following expression:

$(A-(B*C))/(D+(E\hat{}F))$ 3

[b] Explain the following terms with suitable example

(i) Abstract Data Type (ADT) (ii) Triangular Array 2

3[a] Write pseudocode for the followings:

(i) Deletion operation of STACKS

(ii) Insertion operation in a CIRCULAR QUEUE. 3

[b] A 2D Array DATA [6][8] is stored in a column major order with base address 301 and size 1 calculate the address of DATA [2][4]. 2

4[a] How do you model an queue using stack? Illustrate all possible steps. 3

[b] Write a recursive function to count the number of occurrences of the number 24 in an array of n integers. 2

OR

Write the real world applications of STACK & QUEUE data structure.

*Total No. of Pages: 3*

*Roll No. MC/009*

**IIIRD SEMESTER** **B.Tech.(Mathemtics and Computing)**

**END SEMESTER EXAMINATION** **(Nov. – 2013)**

Paper Code: **MC-204** Subject: **Data Structures**

**Time: 3:00 Hours** **Max. Marks: 70**

**Note:** Answer any five questions.
Assume suitable missing data, if any.

1. (a) What is a Binary Search Tree (BST)? Make a BST for the following sequence of numbers.

45, 36, 76, 23, 89, 115, 98, 39, 41, 56, 69, 48

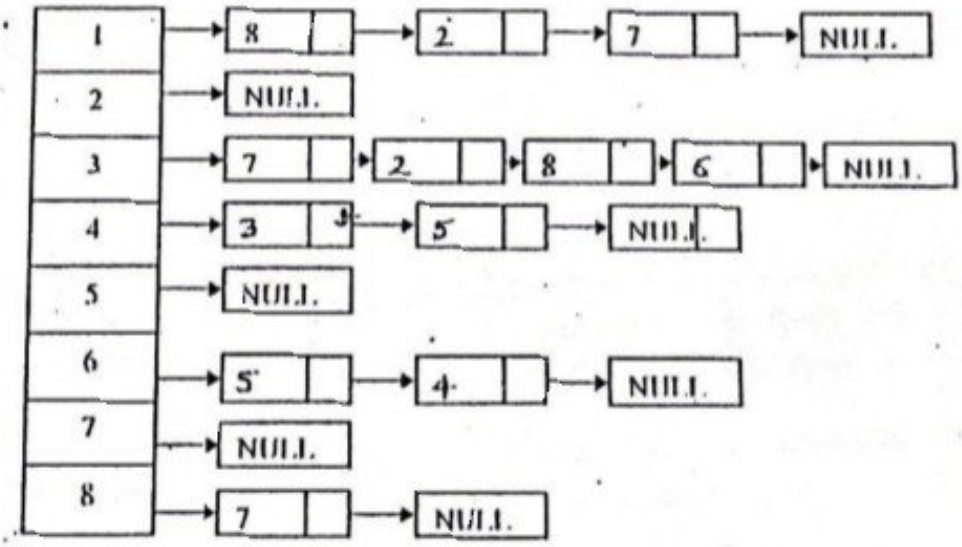
Traverse the tree in Preorder, Inorder and postorder.

**(b)** Draw the expression tree of the following infix expression. Convert it in to Prefix and Postfix expressions.

((a + b) + c * (d + e) + f )* (g + h)

(7+7=14)

2. **(a)** Consider a graph having following representation. Draw the correcponding graph. Run Depth First Search (DFS) on this graph starting from node 3 and show the result.



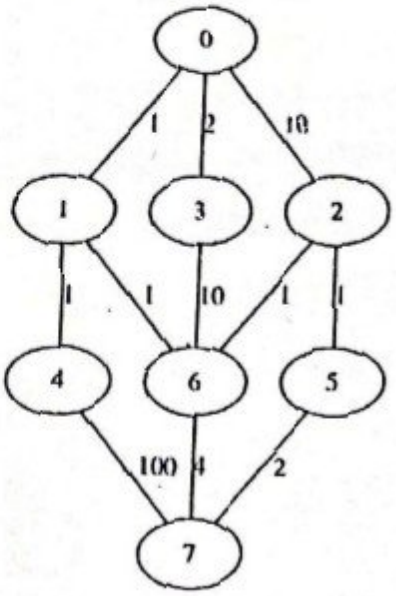

(b) Draw a B-tree of order 3 for the following sequence of keys:

2, 4, 9, 8, 7, 6, 3, 1, 5, 10

(7+7=14)

3. (a) Reverse the order of elements on a stack S
   (i) using two additional stacks.
   (ii) using one additional queue.

(b) Find the minimum spanning tree for the following graph using Prim's algorithm starting from vertex 0. Show every step of algorithm.




(6+8=14)

4. (a) Define the term array. How are two-dimensional arrays represented in memory? Explain how address of an element is calculated in a two dimensional array.

(b) Which sorting algorithm is easily adaptable to singly linked lists? Explain your answer. (7+7=14)

5. (a) Two Binary Trees are similar if they are both empty or if they are both nonempty and left and right sub trees are similar. Write an algorithm to determine if two Binary Trees are similar.

(b) How can stacks be used to check whether an expression is correctly parenthesized or not. For eg(()) is well formed but ({) or )(){ is not.

(7+7=14)

6. (a) Write an algorithm to convert an unsorted array into a min-heap.

(b) Illustrate your algorithm with the following sequence of keys:

8, 20, 9, 4, 15, 10, 7, 22, 3, 12

(c) Write down delete-Min() algorithm and

(d) Show the heap structure after each delete-Min() operation on heap constructed above (perform two such operations).

(3+4+3+4=14)

7. (a) Construct a binary tree whose nodes in inorder and preorder are given as follows:

Inorder : 10, 15, 17, 18, 20, 25, 30, 35, 38, 40, 50

Preorder: 20, 15, 10, 18, 17, 30, 25, 40, 35, 38, 50

(b) Delete the keys "30" from the constructed tree and give the resultant tree.

(7+7=14)

8. Write short notes on following

(i) B Tree.

(ii) Time Complexity, Big O notation.

(iii) Merge Sort.

(iv) Breadth First Traversal. (4x3.5=14)

*Total No. of Pages 1*

*Roll No.* MC/08

3TH SEMESTER

B.Tech.(MC)

MID SEMESTER EXAMINATION

(SEPTEMBER. – 2013)

MC- 204

DATA STRUCTURE

**Time: 1:30 Hours**

**Max. Marks: 20**

**Note:** Answer any 5 questions.

Q1- What is Data structure? Distinguish between Primitive and non primitive data structure. (4)

Q2- Define complexity of an algorithm. Distinguish between time and space complexity? (4)

Q3- Each element of an array X [20][50] requires 4 byte of storage whose first element is stored at location 2000.Determine the location of x[10][10] when array is stored as (4)

i. Row major

ii. Column major

Q4- Define and explain – (4)

i. Big -Oh notation.

ii. Big-omega notation.

Q5- Write Insertion and traversing algorithm for linear array. (4)

Q6- Using insertion sort find out the value after 5th iteration of following array (4)

40,50,12,30,90,18,06,60,

THIRD SEMESTER **B.Tech. (MC)**

**END SEMESTER EXAMINATION** ***(Nov.-Dec.-2012)***

## MC-204 DATA STRUCTURE

***Time: 3:00 Hours*** ***Max. Marks : 70***

**Note :** Answer any *FIVE* questions.
Assume suitable missing data, if any.

1[a] What are the balancing factors of an AVL tree? Consider the following keys inorder and construct an AVL tree. Show your work step by step. 7

74, 2, 14, 25, 12, 100, 98 and 83.

[b] What are the principal criteria for choosing a Hashfucntion? Write different methods for finding the Hashfunction for a particular problem. 7

2[a] What is B-tree? Draw B-tree of order 3 for the following sequence of keys

2,4,9,8,7,6,3,1,5,10

and also write the application of STACKs and QUEUES with examples. 7

[b] Differentiate between strictly binary tree and threaded binary tree and also explain the responsibilities of a file system with suitable example. 7

3[a] Define a linked list. How are these stored in the memory? Suppose the linked list in the memory consisting of numerical values. Write the procedure for each of the following

(i) To find MAX of the values in the list.

(ii) To find the average MEAN of the values in the list. 7

[b] Create a Binary search tree (BST) for the following sequence

45, 32, 90, 34, 68, 72, 15, 24, 30, 66, 11, 50, 10

Traverse the BST created in preorder and post order. 7

4[a] Describe the Dijkstra's algorithm for finding a shortest path in a given graph with suitable example. 7

[b] Construct a binary tree whose nodes in inorder and preorder are given as

Inorder : 10,15,17,18,20,25,30,35,38,40,50

Preorder:20,15,10,18,17,30,25,40,35,38,50 7

5[a] Consider the following array

20,15,22,30,50,2,8,9,10,60

Show the contents of the array after each sort listed below

(i) Insertion sort (after 3rd iteration)

(ii) Quick sort (After 4th iteration)

(iii) selection sort (after 3rd iteration) 7

[b] What is the difference between prims algorithm and Kruskal's algorithm for finding the minimum spanning tree of a graph? and also explain about mirror image of a tree. 7

6[a] Write a program to subtract two polynomials using linked list and also write an algorithm to insert a node in the beginning of the linked list.

[b] Write an algorithm to search a key in a B-tree. What is the worst case of searching in a B-tree? List the possible situations that can occur while inserting a key in a B-tree? 7

7 Write short notes on any FOUR of the following

[a] Abstract data type and asymptotic notation

[b] Collision resolution techniques

[c] File organization

[d] Adjacency matrix of a graph

[e] DFS and BFS

[f] Doubly linked list and priority queues. 4×3.5

-168-

*Total No. of Pages:* 2

Roll No.....

**III[rd] SEMESTER**

**B.TECH. [IT]**

**Supplementary Examination**

**(Feb. - 2019)**

## IT-201 DATA STRUCTURES

**Time: 3:00 Hours** **Max. Marks: 40**

**Note:** Answer any five questions.
Assume suitable missing data, if any.

**Q.1.**

a) Explain the average case complexity of Merge Sort. [3]

b) Explain the procedure to convert infix to postfix expression and evaluate the following postfix expression 7 3 4 + - 2 4 5 / + * 6 / 7 +? [5]

**Q.2.**

a) Write an algorithm to find the 3rd elements from the end in a linked list in one pass. [4]

b) Write the difference between Stack and Queue. Also explain all the operations on Queue data structure. [4]

**Q.3.**

a) Assuming that priority queue is implemented using the linked lists where a master list contains a pointer to the corresponding priority list. Write a function to insert an element x of priority p into this queue. [4]

b) Write a program to implement stack using singly linked list. [4]

**Q.4.**

a) What are the advantages of Complete Binary tree? Explain the operations of Complete Binary tree with suitable example. [4]

**P.T.O**

b) Write an algorithm for heap sort. Discuss its complexity. For th following key sequence determine the binary heap obtained whe the keys are inserted one by one in the order given into an initiall empty heap and perform heap sort: 9, 91, 74,58, 45, 11, 76, 40, 9 15. [4

**Q.5.**

a) How we can represents the graph explain? Write Depth First Searc (DFS) algorithm for traversing a graph. [4

b) What are the advantages of hash tables? Explain with a suitab example. [4

**Q.6.** Write short notes on any four: - [4x

a) Binary Search
b) Threaded trees
c) Collision resolution techniques
d) Spanning Tree
e) External Sorting

**-END-**

*Total No. of Pages 2*

IIIRD SEMESTER

MID SEMESTER EXAMINATION

*Roll No.....*

B.Tech.-(Computer Engg.)

(Sept. – 2019)

Paper Code: COE-201

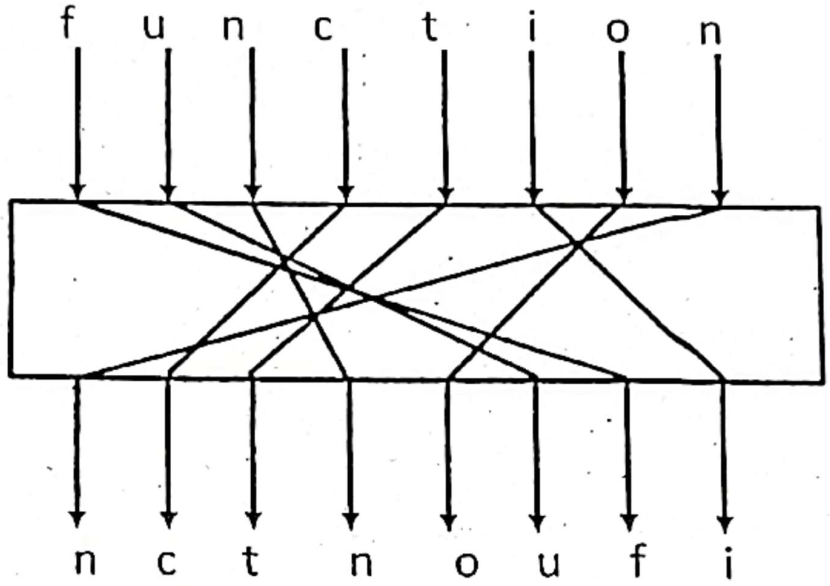
Title of the subject: Data Structures

Time: 1:30 Hours

Max. Marks: 30

Note: Answer all questions.
Assume suitable missing data, if any.

1. Consider following diagram to rearrange characters of 8-char array (e.g. string




"function" is converted to string "nctnoufi".

(a) Design a suitable data structure (say 'X') and write function to perform this conversion.

(b) Write a function to find inverse of data structure 'X' such that string "nctnoufi" is converted back to "function" using this new structure.

(4+4=8 marks)

2. There is a singly linked list containing data sorted in non-decreasing order. There are two elements in this linked list which store similar data values. Write an algorithm to delete first node (out of two-containing equal values) which contains duplicate data value.

(5 marks)

3. Write an algorithm for returning value stored in $k^{th}$ node (from end) of the singly linked list.

(5 marks)

4. Use a stack to test for balanced parentheses, when scanning the following expressions. Only consider the parentheses [,],(,),{,} . Ignore the variables and operators. Example inputs (valid balanced) are:
   (a) [ a + { b / ( c - d ) + e / (f + g ) } - h ]
   (b) [ a { b + [ c ( d + e ) - f ] + g }
   Write algorithm to test if given input string contains balanced parentheses.

(6 marks)

5. Let S be a stack of size n ≥ 1. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and pop operation take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation.
   (a) For m ≥ 1, define the stack-life of m as the time elapsed from the end of Push(m) to the start of the pop operation that removes m from S.
   (b) Compute the average stack-life of an element of this stack.

(6 marks)

*Total No. of Pages: 2*

Roll No.....

**III[rd] SEMESTER**

**B.TECH. [IT]**

**Supplementary Examination**

**(Feb. - 2018)**

**IT-201 DATA STRUCTURES**

**Time: 3:00 Hours** **Max. Marks: 40**

**Note:** Answer any five questions.
Assume suitable missing data, if any.

**Q.1.**

a) Explain the worst case complexity of Bobble sort. [2]

b) Reorder the following complexity from largest to smallest $2^n$, n!, $n^{10}$,$2^4$, $n\log_2(n)$. Justify your answer. [2]

c) Explain the procedure to evaluate postfix expression. Evaluate the following postfix expression 7 3 4 + - 2 4 5 / + * 6 / 7 +? [4]

**Q.2.**

a) Write an algorithm to implement doubly linked list and verify it with suitable example. [4]

b) Write the difference between Stack and Queue. Also explain all the operations on Stack data structure. [4]

**Q.3.**

a) Assuming that priority queue is implemented using the linked lists where a master list contains a pointer to the corresponding priority list. Write a function to insert an element x of priority p into this queue. [4]

b) Write a program that to find middle element of linked list in one pass. [4]

**P.T.O.**

**Q.4.**

a) What are the advantages of AVL tree? Explain the operations of AVL tree with suitable example. [4]

b) Write an algorithm for heap sort. Discuss its complexity. For the following key sequence determine the binary heap obtained when the keys are inserted one by one in the order given into an initially empty heap and perform heap sort: 9, , 91, 74,58, 45, 11, 76, 40, 98, 15. [4]

**Q.5.**

a) Explain various techniques to represent the Graph in computer memory. Write Breath first Search Traversal (BFS) algorithm for traversing a graph and also explain it with an example. [4]

b) What are the advantages of hash tables? Explain with a suitable example. [4]

**Q.6.** Write short notes on any four: - [4x2]

a) Priority Queues using heaps
b) Minimum spanning trees
c) Topological Sorting
d) Doubly Circular Linked List
e) Collision resolution techniques

**-END-**

*Total No. of Pages:* 2 Roll No.....

**III RD SEMESTER** **B.TECH. [IT]**

**END SEMESTER EXAMINATION** **(NOVEMBER - 2019)**

**IT201** DATA STRUCTURES

**Time: 3:00 Hours** **Max. Marks: 40**

**Note:** Answer any five questions.
Assume suitable missing data, if any.

1)

a) Consider an array A[1:1000] with numbers from 1 to 999 in a random order. Except one number every other number is unique. Write an algorithm to find the duplicate number without accessing the array elements more than once. (4)

b) Differentiate between arrays and linked lists. (2)

c) Why is recursion considered an application of stacks? (2)

2)

a) Given a pattern containing only I's and D's. **I** for increasing and **D** for decreasing. Write a program using stacks to print the **"minimum number"** for any such pattern. Also dry run the program for DDIDDIID. Digits from 1-9 and digits can't repeat. (5)

Example patterns are:-

| | |
|---|---|
| Input: D | Output: 21 |
| Input: I | Output: 12 |
| Input: DD | Output: 321 |
| Input: II | Output: 123 |
| Input: IDI | Output: 1324 |
| Input: DIDI | Output: 21435 |
| Input: IIDDD | Output: 126543 |
| Input: DDIDDIID | Output: ? |

b) Write an algorithm for merging two linked list. (3)

3)

a) Consider a system which implements only a queue data structure. Write a program to support the normal operations such as insertion and deletion in a stack using queue operations. Use minimum number of queues. (4)

b) Write a program to split a stack into two, with the first stack having the bottom half elements and the second stack having the remaining

elements. For example, if the stack has elements 1, 2, 3, 4, 5 (in order) with top pointing to 5, the split stacks must be 1,2 and 3,4,5 with top pointing to 2 and 5, respectively. (4)

4)

a) Write a program in C/C++ to find the level in a binary tree with the maximum number of nodes and return the number of nodes. (4)

b) What is the need for height balanced trees? Who introduced them? Insert the following keys in a Binary search tree so that it is a AVL tree after each insertion. (4)

32, 45, 13, 4, 26, 10, 120, 99, 88, 64, 12

5)

a) Make a B-tree of order 3 by inserting the following keys to an empty 3-way B-tree. (4)

65, 8, 15, 3, 9, 16, 5, 13, 18, 33, 28, 60

b) Write an algorithm for heap sort. Discuss its complexity. For the following key sequence determine the binary heap obtained when the keys are inserted one by one in the order given into an initially empty heap and perform heap sort.

42, 91, 58, 66, 70, 12, 16, 75, 40, 39, 11 (4)

6)

a) Explain various techniques to represent the Graph in computer memory. Write Depth first Traversal (DFS) algorithm for traversing a graph and also explain it with an example. (4)

b) Given the keys {329, 410, 333, 970, 280, 159, 865, 593, 621, 564} and the mid square hash function, show the contents of the hash table of size 10 which resolves collision using linear probing. (4)

7) Write short notes on any four: - (4x2)

a) Collision resolution techniques
b) Topological Sorting
c) Right and left Threaded Binary Trees
d) Doubly Circular Linked List
e) Priority Queues

**-END-**

*Total No. of Pages:2*

**IIIrd SEMESTER**

**MID SEMESTER EXAMINATION**

*Roll No.....*

**B.Tech(SE)**

**(Sept. – 2018)**

Paper Code: **SE-201**

Time: 1:30 Hours

**Title: Data Structures**

Max. Marks: 30

**Note:** 1. Attempt all questions. Assume any suitable value(s) for missing data.
2. If asked to write algorithms, write as C functions or in pseudo code.

1. (a) Create a data structure *twoStacks* that represents two stacks. A single array A[1..MAXSIZE] is used to implement *twoStacks*. The two stacks S1 and S2 grow from opposite ends of the array. Variables top1 and top2 (topl< top 2) point to the location of the topmost element in each of the stacks. Following functions must be supported by *twoStacks*.

Push1(int x) –> pushes x to first stack
push2(int x) –> pushes x to second stack

pop1() –> pops an element from first stack and return the popped element
pop2() –> pops an element from second stack and return the popped element

Write the algorithm for the Implementation of above said *twoStack*. If the space is to be used efficiently, what should be the condition for "stack full"? [6+2=8]

(b) Find out the time and space complexity of following C- program [2]

```
int i , j, k, sum=0;
for (i=0:i<n; i++)
  for(j=0;j<n; j++)
    for(k=0;k<9999;k++)
      sum++;
```

2. (a) Assume a set is represented by a linked list. Write down a best algorithm to perform union and intersection operation of two sets each represented by linked list containing n elements and find out the time complexity of each operation. [2+2+2=6]

(b) The following C function takes a singly linked list of integers as a parameter and rearranges the elements of the list. The list is represented as pointer to structure. The function is called with the list containing integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes?

```
struct node{
        int value;
        struct node *next;
        };
void rearrange(struct node *list)
  {
```

P.T.O

```
struct node *p, *q; int
 temp;
if (list==NULL || list → next==NULL) return;
p = list;
q = list → next;
while(q!=NULL)
{ temp = p → value;
p → value = q → value;
q → value = temp;
p = q → next;
if (p!=NULL) q=p → next;
    else q= NULL;
}
}
```

[4]

3. (a) Perform the following operations using stack:
   Postfix to Prefix expression: AB+CD+*E/
   Evaluate the following Postfix expression: 2 4 6 * - 9 8 3 / + * 5 ^ 2 +
   Where ^ is an exponential. [2+2=4]

   (b) Why circular queue is preferred over linear queue? Write down the underflow, overflow condition of Circular queue? Write down the algorithm for traversal of circular queue considering all cases. [1+1+1+1]

   (c) Define the following terms in tree data structure: [2]

   - Depth of a node
   - Height of a node

**END**

*Total No. of Pages:2*
**IIIrd SEMESTER**
**MID TERM EXAMINATION**
Paper Code: **SE-201**

*Roll No.....*
**B.Tech (SE)**
**(Sept – 2019)**
**Title: Data Structures**

Time: 1:30 Hours
Max. Marks: 30

**Note:** 1. All questions are compulsory.
2. Assume any suitable value(s) for missing data.
3. If asked to write algorithms, write as C/C++ functions or in pseudo code.

Q 1. **(2*5=10 marks)**

(a) What is meant by stack overflow condition? Is it applicable to the linked list method of implementation of the stack? Give reasons.

(b) Consider a two-dimensional array A [20][10]. Assume 4 words per memory cell, the base address of array A is 100, elements are stored in row-major order and first element is A[0][0]. What is the address of A[11][5] ?

(c) What is the output of following function for start pointing to first node of following linked list? 1->2->3->4->5->6

```
void fun(struct node*start)
{
 if(start == NULL)
  return;
 printf("%d  ", start->data);
 if(start->next != NULL )
  fun(start->next->next);
 printf("%d  ", start>data);
}
```

(d) What is the time complexity of the function?

```
int fun(int n)
{
 int count = 0;
 for (int i = n; i > 0; i /= 2)
   for (int j = 0; j < i; j++)
     count += 1;
 return count;
}
```

(e) The result of evaluating the postfix expression 10, 5, +, 60, 6, /, *, 8, – is

Q 2.

a) Consider the following queue of characters where QUEUE is a circular array which is allocated six memory cells: **(3 marks)**
Front=2 Rear=4 QUEUE: _, A, C, D, _, _ ("_" denotes empty memory cells).
Describe the queue after each step as the following operations takes place:
i) F is added to the queue, ii) Two letters are deleted, iii) K, L, and M are added to queue, iv)Two letters are deleted, v) R is added to queue, vi) Two letters are deleted, vii) S is added to queue, viii) Two letters are deleted.

b) Translate the following infix expression into postfix expression: **(2 marks)**
A + (B * C – (D / E / F) * G) * H

c) Let A and B be two linked list. Write a 'C' function to create a new linked list C that contains elements alternately from A and B beginning with the first element of A. If you run out of elements in one of the lists, the append the remaining elements of the other list to C. **(5 marks)**

Q 3.

a) Write an algorithm to concatenate two singly linked list? **(3 marks)**

b) Create a data structure *twoStacks* that represents two stacks. A single array A[1...MAXSIZE] is used to implement *twoStacks*. The two stacks S1 and S2 grow from opposite ends of the array. Variables top1 and top2 (top1<top2) point to the location of the topmost element in each of the stacks. Following functions must be supported by *twoStacks*.
Push1(int x)→ pushes x to first stack.
Push2(int x)→ pushes x to second stack.
Pop1()→ pops an element from first stack and return the popped element.
Pop2()→ pops an element from second stack and return the popped element.
Write an algorithm for the implementation of above said *twoStack*. **(5 marks)**

c) Write an algorithm to insert a new node 'p' in a doubly linked list after a node pointed by a node pointer 'q' consider all the cases. **(2 marks)**

*Total No. of Pages02* *Roll No. .....*

THIRD SEMESTER **B.Tech. ( IT )**

**MID SEMESTER EXAMINATION** ***SEPT-2019***

**IT-201 DATA STRUCTURES**

*Time: 1:30 Hours* *Max. Marks : 30*

**Note :** Attempt all Questions.
Assume suitable missing data, if any.

1)

a) Classify different types of Data structures. Also discuss how they can help in reducing space complexity for different real world problems. (3)

b) Give the sequence of argument values that result when the following program is invoked for each of integers 3, 4 and 5. (4)

```
int puzzle(int n)
{
    If (n==1)
        return 1;
    If (n%2==0)
        return puzzle(n/2);
    else
        return puzzle(3*n+1);
}
```

2)

a) What is the output of the following code snippet? (3)

```
#include <stdio.h>
int main()
{
    int arr[][3]={{1,2},{3,4,5},{5}};
    printf("%d%d%d", sizeof(arr), arr[0][2], arr[1][2]);
    return 0;
}
```

b) Give an algorithm to convert a Prefix expression to an Infix expression using Stacks. (5)

3)

a) Write a program to insert a node with info 5 after each node having the info as 4 in a linked list. (4)

b) Implement a stack using arrays in C. (4)

4)

a) Write a program to delete redundant elements from a queue. (4)

b) Give the linked implementation of a queue and discuss the implementation of priority queues. (3)

**-End-**