Department of Electrical Engineering
Assignment
Date: 25/09/2020

Course Details

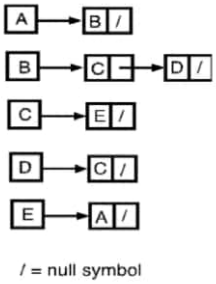
Course Title: Data Structure and Algorithm **Module:** _____
Instructor: _____ **Total Marks:** 50

Student Details

Name: _____ **Student ID:** _____

Note: Plagiarism of more than 20% will result in negative marking.
Similar answers of students will result in cancellation of the answer for all parties.

Q1.	Explain the following operations in a single link list. 1. Insert an Element 2. Delete an Element	CLO 1 Marks 10
Q2.	Write a program to insert a new element in the given unsorted array at K^{th} position	CLO 2 Marks 12
Q3.	Explain Quick sort with the help of suitable example Write an algorithm for insertion in sorted linked list	CLO 1 Marks 13
Q4.	e adjacency list representation of a graph with f rtices A, B, C, D, E is given below.	CLO Marks 15



/ = null symbol

Part (a). Draw the adjacency matrix (complete the partial drawing below), and

Naveed Ali

ID NO: 16753

Q1 Explain the following operation in a ~~sign~~ single link list.

1) Insert an Element.
2) Delete an Element.

Ans:

Single Link list is a sequence of element in which every element has link to its next element in sequence.

In a ~~sign~~ single linked list the inserting operatio can be performed in three ways.

1. Inserting at begining of the list.

NULL)

Steps 1.

cerate a newnode with the given value.

p2 Step 2.

Check whether list is empty (head == NULL).

Step 3.

If H is Empty then set newNode → next = NULL & head = New node

Step 4.

If it is not Empty then, Set new node → next = head & head = NewNode.

2. Inserting at End of the list

Step 1

Create a new node with the given value & newNode → next as NULL.

Step 2.

Check wheater list is Empty Cheeel == NUL.

Step 3.

If it is Empty then Set. new Node → next = NULL & head = New Node.

3 Inserting at the End of List.

Step 1

Creat a new Node with given value & new node → next as NULL.

Step 2.

Check wheeter list is me Empty (head == NULL).

Step 3.

If it is Empty than, set head = New Node.

Step 4.

If it is not Empty than depire a node pointer temp & initialize with head.

(ii) Delete an Element.

In a single linked list. The deletion operation can be performed in three ways.

1. Deleting from Begining of the list.

Step 1.

Check wheeter list is Empty (head == NULL)

Step 2.

If it is Empty than, display List is empty !!! Deletion is not

possible' and terminate the function.

Step 3.
If it is not empt then, define a Node pointer 'temp' & initialize with head.

Step 4. Check wheater list having only one node (temp → next == NULL)

Step 5.
If it is true than set head = NULL & delete temp.

2. Delete from End of the list.

Step 1
Check wheater list is Empty (head == NULL)

Step 2.
If it is not empty than define two node pointers (temp1) and temp2.

Step 3.
Check list has only one Node ~~(temp1' with head.~~
(temp1 → next == NULL.

Step 4.

if it is ture. then set head = NULL & delete.

3) Deleting a specific Node from list.

Step 1

check. list is empt. (head == NULL)

Step 2.

if Not empty then define two Nodes. temp1 & temp2. & Intidize temp1. with head.

— x — x — x — x — x — x —

Q3: Explain Quick sort with help of suitable example.

Quick sort:

Quick Sort is a divide & conquer algorithm. It picks an element as pivot & partits the given array around the picked pivot. There are many different versions of quick sort that pick pivot in different ways

1) Alway pick 1st element as pivit.

2) Alway pick last element as pivit.

3) Pick a random element as pivat.

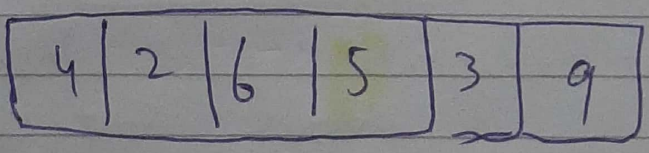
4) pick median as pivot.

The key process in quick sort is partition ().

Target of partition is given an array and an element x of array as pivit. Put x at its correct position in sorted array & put all smaler element. (Smaller than x). before x. and put all ~~geal~~ greater element (greater than x). After x. All this should be done in Liner time.

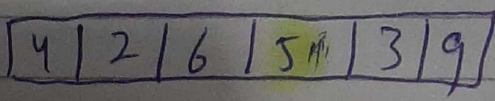
~~simple~~.

Example.

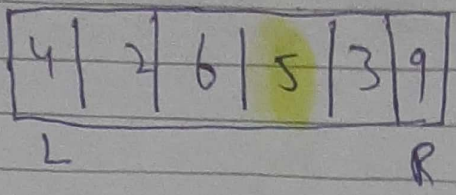


We will use Quick Sort.

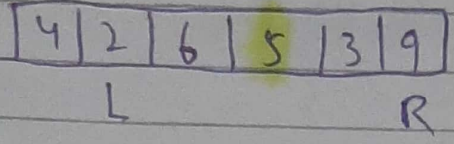
Step 1



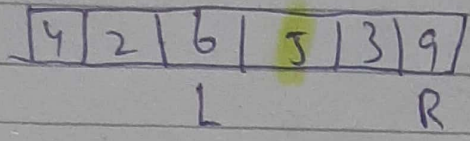
Step 2.



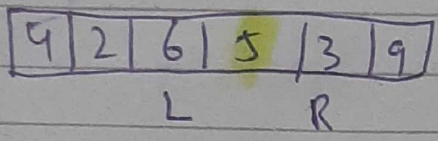
Step 3

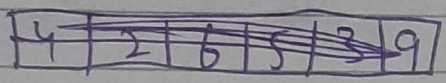


Step 4

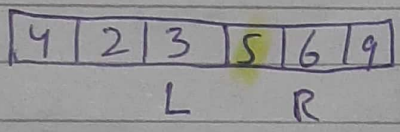


Step 5

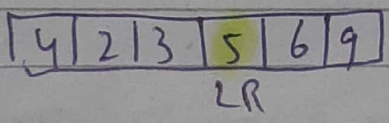




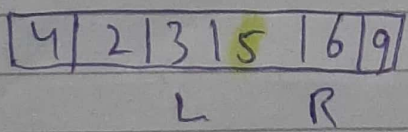
Step 6



Step 7.

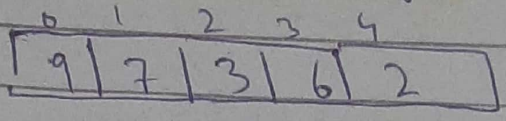


Step 8.



Q3 (b) Algorithm for insertion sorted Link list

Insertion Sort Algorithm



a: array of items
n: size of list

Start
1. Declare variables - i, key, j
2. loop : i to n-1 // outer loop
2.1 key = a[i];
2.3 Loop: (j > 0 && array[j] > key)
// inner loop
2.3.1 arr[j+1] = arr[j];
2.3.2 j = j-1;
end loop // inner loop
2.4. arry[j+1] = key;
end loop. // outer loop
Stop.

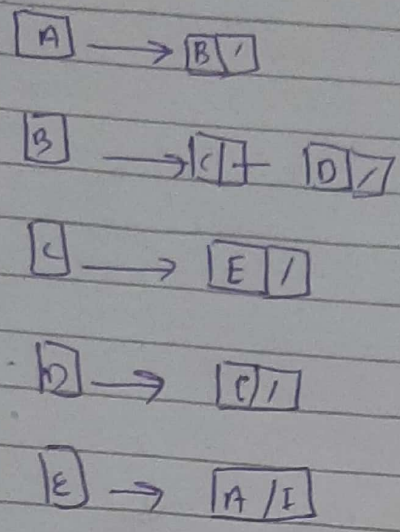
Q2:- Write a program to insert a new element in given unsorted array at K^{th} position.

Ans:

```
#include <stdio.h>
int main()
{
    int arr[100] = [0]
    int i. K pos, n = 10;
    // initial array of size 10
    for (i = 0; i < 10; i++)
        arr[i] = i + 1.
    // print the original array
    for (i = 0; i < n; i++).
        printf ("%d" arr[i]);
    print f ("/n");
```

```
// element to be inserted
    k = 50;

// position at which element
// is to inserted

    pos = 5;
// increase the size by 1
    n++;

// shift element forword

for (i = n; i >= pos; i--)

    arr[i] = arr[i-1];

// insert k at pos

    arr[pos-1] = k;

// print the upated array

for (i = 0; i < n; i++)

printf ("%d", arr[i]);

printf ("\n");

return 0;

}
```

Q4: The adjacency list representation of a graph with five vertices A, B, C, D, E is given below



A)

Adjacency matrix is given below for directed graph.

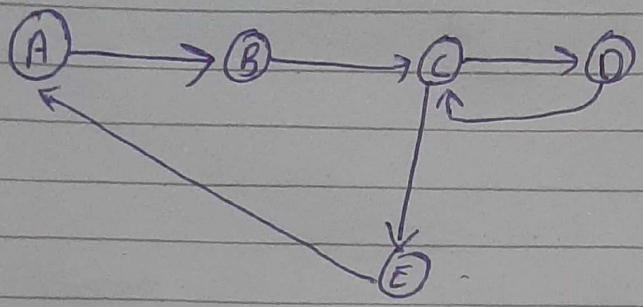
(a)

	A	B	C	D	E
A	0	1	0	0	0
B	0	0	1	0	0
C	0	0	0	1	1
D	0	0	0	1	0
E	1	0	0	0	0

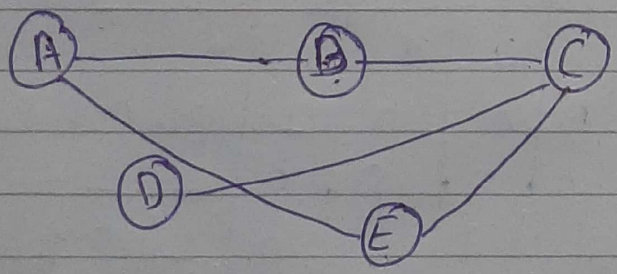
B) Draw the graph.

A Graph (G) has two section, the vertices & edges set V and edges are represented as set E. so the graph not is G (V, E)

Now graph for given adjacency List is



For ~~undirectic~~ Undirected graph



A	A	B	C	D	E
B	1	1	0	0	1
C	1	1	1	0	0
D	0	1	0	1	1
E	1	0	1	0	0

	A	B	C	D	E
A	0	1	0	0	1
B	1	0	1	0	0
C	0	1	0	1	1
D	0	0	1	0	0
E	1	0	1	0	0