



*Name : Muhammad Musa*

*Department : BS(CS)*

*Semester : 4th*

*ID # : 15366*

*Paper Subject : Design and Analysis  
of Algorithm*

*Submitted To : Muhammad Adil Sir*

*Dated : 14<sup>th</sup> April 2020*

①

BS(cs) Muhammad Musa, ID# 15366, Sir M. Adil  
Design and Analysis of Algorithms

Q.1:-

Ans: (a)

### Linked List:-

A linked list is a list whose elements may not occupy continuous memory locations and whose elements are connected by means of links between them.

(\*) Each element of a linked list is called node.

(\*) Each node has at least two fields/Parts:

#### 1. Info field:

Info field keeps data.

#### 2. Link field:

Link field keeps address of next node.

• Link field of last node is kept  $\phi$ .

### Head:

A pointer "head" is used to keep the address of 1<sup>st</sup> node.

### Types of Linked List:

There are three types of Linked List.

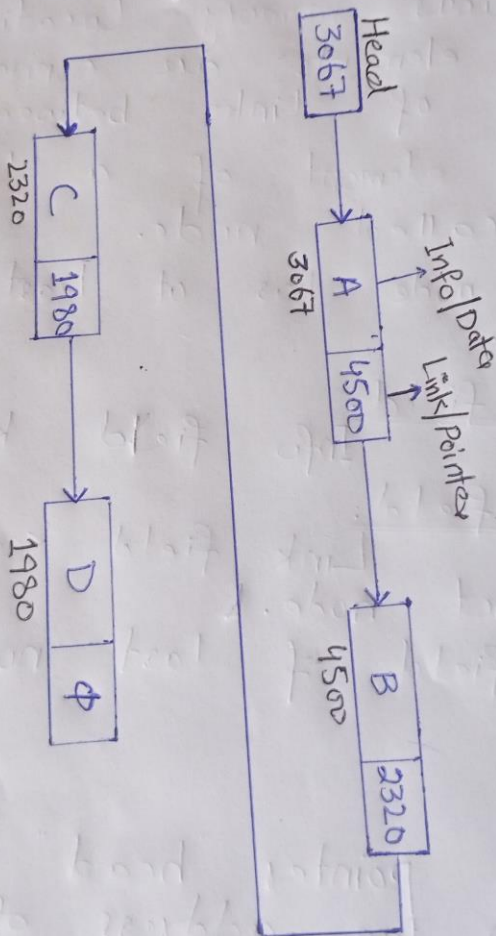
②

BS(CS) Muhammad Musa, ID # 15366

1. One way Linked List.
2. Two way Linked List.
3. Circular Linked List.

Q1:

Ans: (b) Diagrammatic One way Linked List:



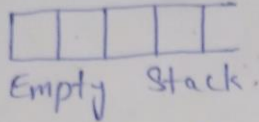


Q2-

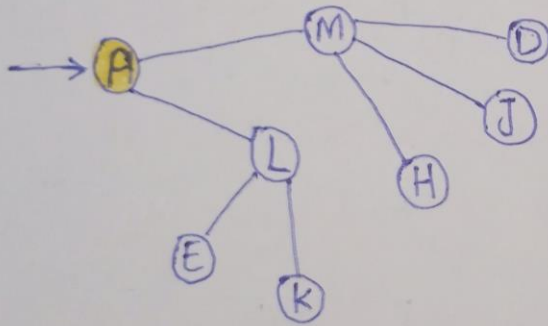
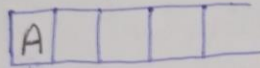
Ans: Depth-first Technique:

Solution:

First we take an empty stack;



- Start from root node "A".
  - Highlight this node "A".
  - Now we push "A" into stack.



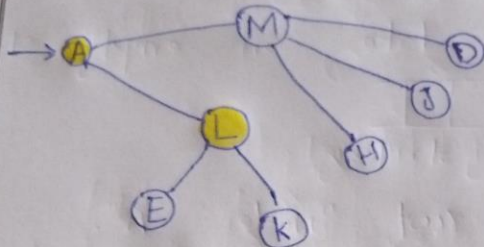
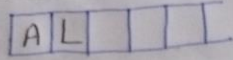
Output Sequence:

A,

- Now "A" is adjacent to "M" and "L".
  - We follow alphabetically, we select "L".
  - Highlight this node "L".
  - Now we push "L" onto the top of stack.

4

BS (CS) Muhammad Musa ID # 15366

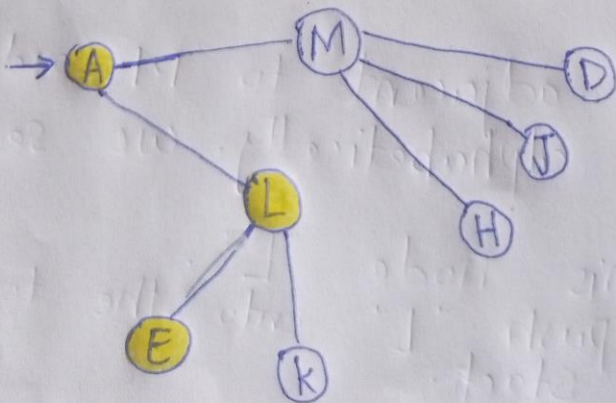
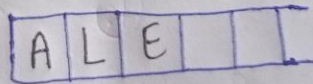


Output Sequence:

A, L

③ Now "L" is adjacent to "E" and "K".

- We follow alphabetically, we select "E".
- Highlight this node "E".
- Now we push "E" on the top of the stack.

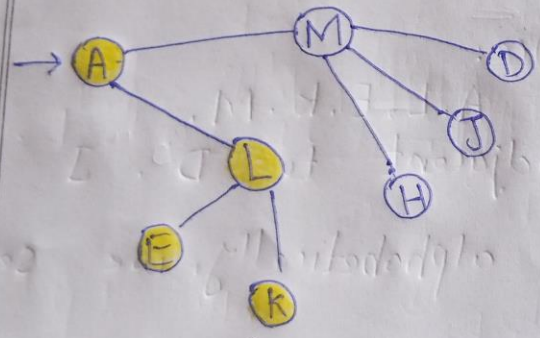
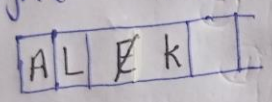


Output Sequence:

A, L, E,

④ As "E" is leaf, so we pop it from stack.

- We get back to "L".
- Now we push "k" on top of the stack.
- Highlight node "k".



Output Sequence:

A, L, E, k,

⑤ "k" is also a leaf, so we pop it from stack.

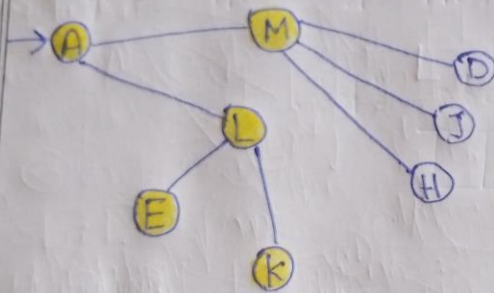
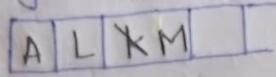
- We get back to "L".
- As "L" has no other adjacent elements, which we are pushed, so we get back to "A".



BS(CS)

Muhammad Musa ID # 15366

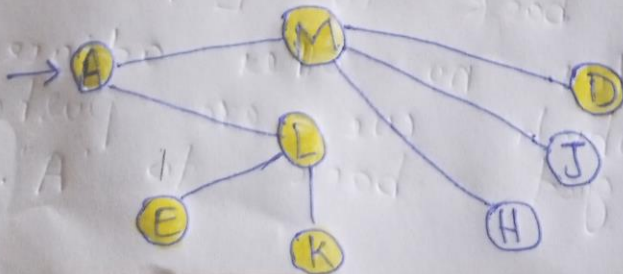
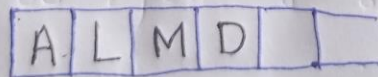
- We push "M" on top of the stack.
- Highlight this node "M".



Output Sequence:

A, L, E, K, M,

- "M" is adjacent to "D", "J" and "H".
- We follow alphabetically, we select "D".
- We push "D" on the top of the stack.
- Highlight this node "D".



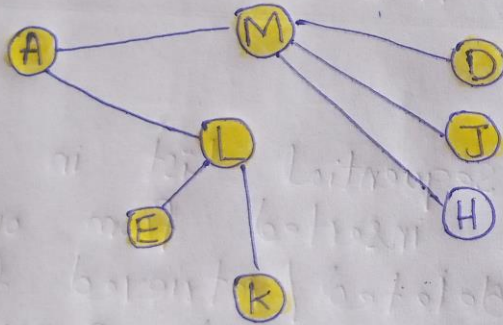
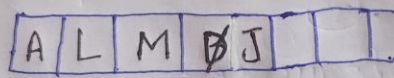
⑦

BS (CS) Muhammad Musa, ID # 15366

Output Sequence:

A, L, E, K, M, D,

- ⑦ As "D" is leaf, so we pop it from stack.
- We get back to "M".
  - Now we push "J" on top of the stack.
  - Highlight this node "J".



Output Sequence:

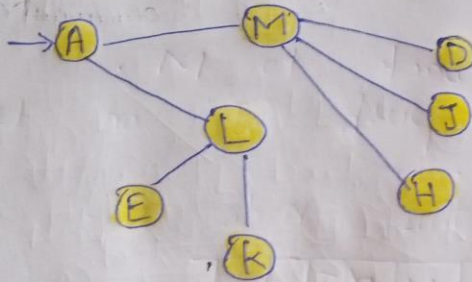
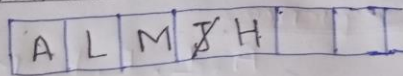
A, L, E, K, M, D, J,

- ⑧ "J" is also leaf, so we pop it from stack.
- We get back to "M".
  - Now we push "H" on top of the stack.
  - Highlight this node "H".



BS (CS)

Muhammad Musa, ID # 15366



Output Sequence:

A, L, E, K, M, D, J, H.

Q3:-

Ans:

Queue:-

A sequential list in which elements are inserted from one end and are deleted/retrieved from other end is called queue.

Rear:-

The end from where an element can be inserted is called rear of the Queue.

Front:-

The end from where an element can be deleted/retrieved is called front of the queue.

(9)

BS(ICS) Muhammad Musa, ID # 15366

Working Principal;

The working principal of a queue is;

"First In - first out" or

"Last In - Last out".

Memory Representation:-

(\*) A linear array  $q[]$  is used to represent a queue.

(\*) Two variables "F" and "R" are used to denote Front and Rear of  $q[]$ .

Examples:

(\*) Automobiles waiting to pass through a signal make up a queue.

(\*) People waiting to submit bills at a bank's window.

(\*) Patients waiting outside the doctor's clinic.

(\*) Luggage checks by luggage checking machine.

{ The END }