

Date: 13/4/2020

(1)

BSCCS) Syed Danish Ali T.D=14712 · Sir M.A.111  
Design and Analysis of Algorithm

Q.1 (a) Ans:-

• 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.

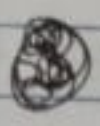
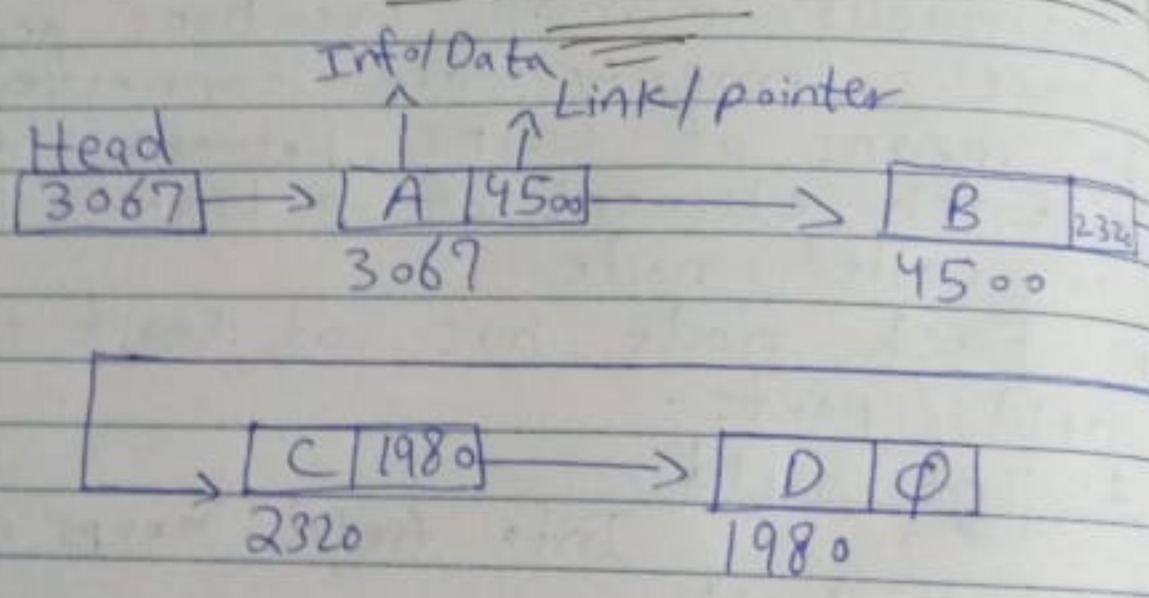
Nov 13/4/2020

(2)

BSCes) Syed Danish Ali ID = 14712

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

Q.1 (b) Ans:- Diagrammatic One way  
Linked List:-

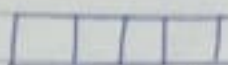


BSCCS) Syed Danish Ali . Id = 14712

Q2. Ans:- Depth-first Technique:-

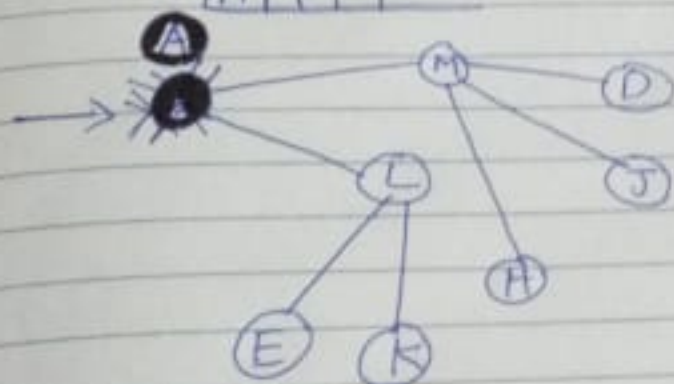
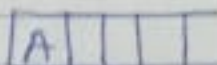
Solution:-

First we take an empty stack



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".

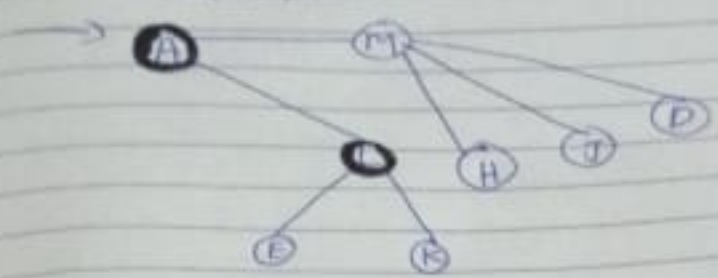
We follow alphabetically we select "L".

Highlight this node "L".

Now we push "L" onto the top of stack.

BSECS) Syed Danish Ali I.D=14712

A L | | |

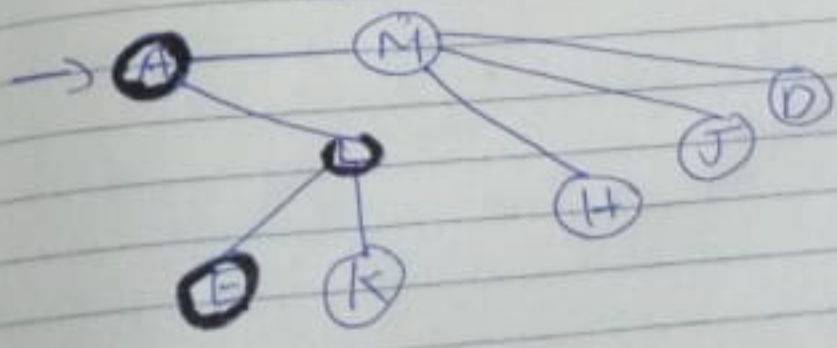


• Output Sequence:  
A, L

③ Now 'L' is adjacent to 'E' and 'K'.

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

A L E | |



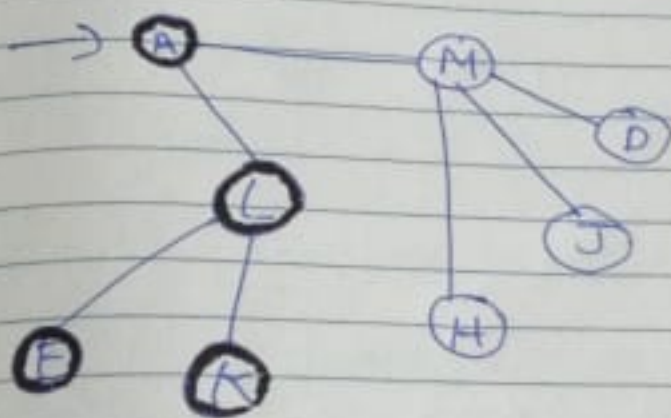
(5)

BSCCS) Syed Danish Ali I.D = 14712

• Output Sequence:-

- 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'.

A | L | ~~E~~ | K |



• Output Sequence:-

A, L, E, K.

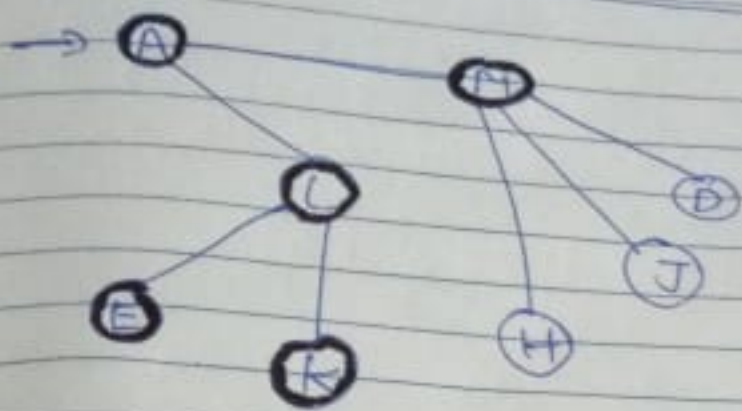
- ⑤ 'K' is also a leaf, so we pop it from stack.
- We get back to 'L'.
- As 'L' has no other adjacent elements, which we are pushed so we get back to 'A'.

6

Date: / /

- BSCCS) Syed Danish Ali I.D = 14712.
- We push "M", on top of the stack.
  - Highlight this node "M".

A L L M I I



• Output Sequence:-

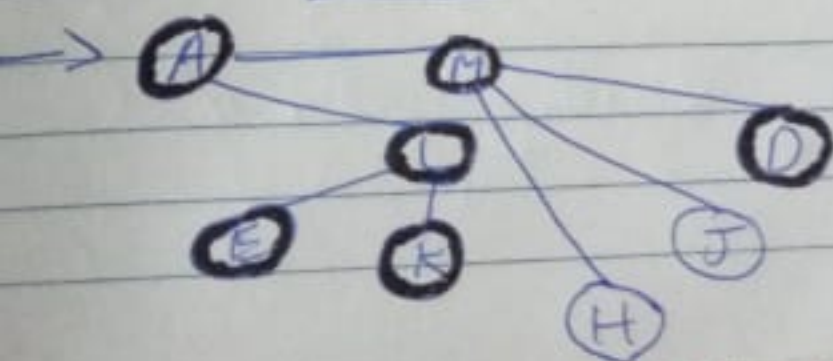
(b) "M" is adjacent to "D", "J" and "H".

• We follow alphabetically, we select "D".

• We push "D" on the top of the stack.

• Highlight this node "D".

A L I M D I I



(7)

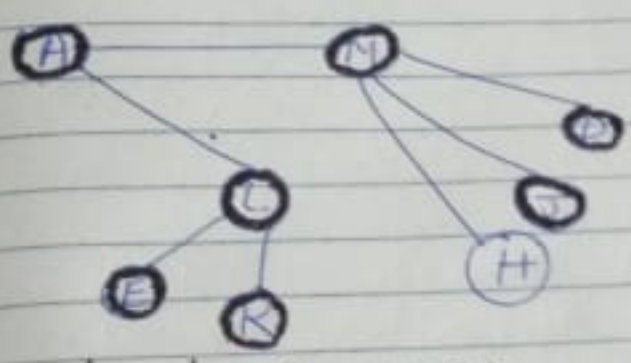
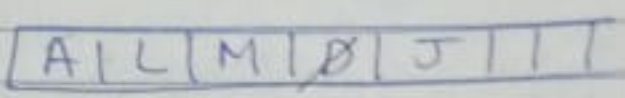
BSCCS) Syed Danish Ali I.D=14712.

• 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'

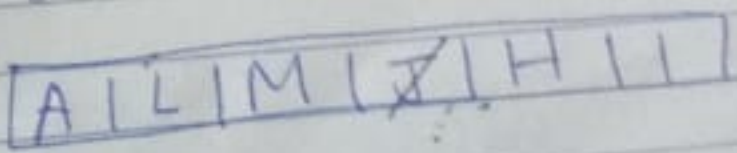


• Output sequence:-

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

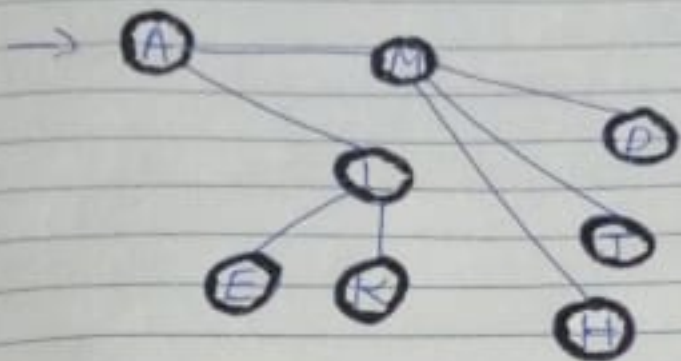
⑧ 'J' is also leaf, so we pop it from stack.

- We get back to 'M'
- Now we push 'H' on top of the stack.
- Highlight this node 'H'



(8)

A I L M J H I I



• Output Sequence:

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

Q.3. Ans:- Queue:

A sequential list in which elements are inserted from one end and are deleted from the ~~other~~ 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)

BSCCS) Syed Danish Ali I.D = 14712

• 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