

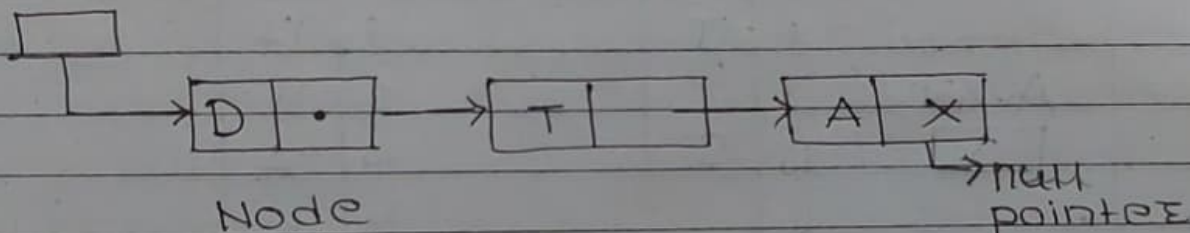
3. Linked List

Page No. _____

Date: ____/____/____

A linked List or one way list is linear collection of data elements called nodes. where each node is divided into two parts first part contain the information of the element and second part contain the address of next node (LINK). The pointer of last node contain special value DATA called as null pointer.

Start



- Representation of Link List in memory

e.g. 1) LINKED LIST

START	INFO	LINK	LINK
1	L	2	2
2	I	3	3
3	N	4	4
4	K	5	5
5	E	6	6
6	D	7	7
7	L	8	8
8	I	9	9
9	S	10	10
10	T	11	11

e.g. 2 STRUCTURES

	INFO	LINK
Start	1 C	1 8
<u>4</u>	2 R	2 5
	3	3 9
	4 S	4 7
	5 U	5 1
	6 E	6 10
	7 T	7 2
	8 T	8 12
Avail	9	9 11
<u>3</u>	10 S	10 -1
	11	11 13
	12 U	12 14
	13	13 15
	14 R	14 6
	15	15 -1

e.g. 3 MARKS OF 1000 SUBJECTS

	DS	DIS
1	16	25
2	20	30
3	15	15
4	30	20
5	25	10

	Test	LINK
DS	1	15
	2	15
	3	.
	4	20
	5	30
DIS	6	
	7	10
	8	
	9	25
	10	30
Avail	11	
	12	20
	13	25
	14	10

for print the element of Linked list.

Traversing a Linked LIST

Let, LIST be linked list in memory. this algorithm traverse list applying and operation PROCESS to each element of list. The variable PTR point to the node currently being processes

PTR = Link [START]

- 1] Set PTR = START
- 2] Repeat Step 3 & 4 While PTR \neq NULL
- 3] Apply PROCESS to INFO [PTR]
- 4] Set PTR = LINK [PTR]. 7/2/5 8/2/7

for 5) Exit.

* Searching

i) Unsorted [Unsorted]

e.g :- 10, 30, 20, 50, 40

Start	TINFO	LINK
<u>4</u>	1 20	1 6
	2	2
	3 40	3 -1
	4 10	4 8
	5	5
	6 50	6 3
	7	7
	8 30	8 1

• Algorithm for Unsorted Searching

SEARCH (TINFO, LINK, ITEM, START, LOC)

1. Set PTR := START
2. Repeat steps 3 While PTR \neq Null
3. IF ITEM \neq TINFO [PTR] then
Set LOC = PTR and Exit
Else
Set PTR := LINK [PTR]
4. Set LOC := Null
5. Exit

ii) Sorted.

e.g 10, 20, 30, 40, 50

START		INFO		LINK
7	1	30	1	5
	2		2	
	3	20	3	1
	4		4	
	5	40	5	8
	6		6	
	7	10	7	3
	8	50	8	-1

SRCHSL (INFO, LINK, ITEM, START, Loc)

1. Set PTR := START
2. Repeat step 3 while PTR \neq Null
3. IF ITEM > INFO [PTR] then
SET PTR := LINK [PTR]
- else IF ITEM = INFO [PTR] then
Set Loc := PTR and Exit
- else Loc := Null and Exit
4. set Loc := Null
5. Exit.

• Header Linked List

A Header link list is a linked list which always contain a special node called as Header node at the beginning of the list.

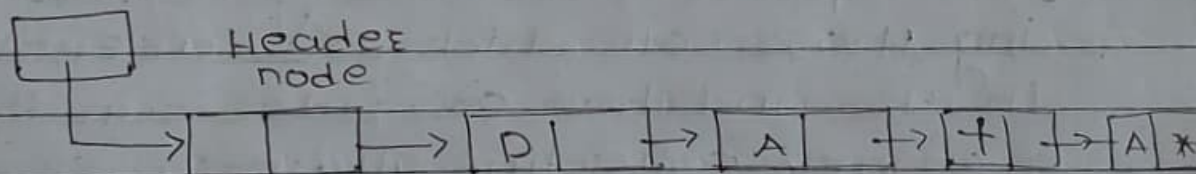
The following are two kind of widely used header list.

(One way list algorithm is grounded header list)

i) Grounded Header list

A grounded header list is a header list where the last node contain the null pointer. (

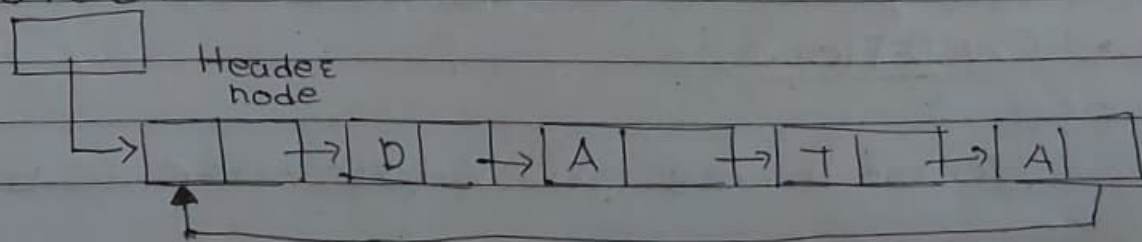
Start



ii) Circular Header list

A Circular header list is a header list where last node point back to the Header node.

Start



- 1) Set PTR := LINK [START]
- 2) Repeat Steps 3 & 4 while
PTR \neq START
- 3) Write INFO

mp • Garbage collection :-

The operating system of computer may periodically collect all the deleted space onto a free storage list. Any technique which does this collection is called as Garbage collection.

Garbage collection usually takes place into two steps

- i) Computer run through all list, tagging those cell which are currently in use and then computer run through memory collecting all untagged space onto a free storage list.
- ii) Garbage collection may take place when there is only some minimum amount of space or no space at all left in free storage list or the CPU is idle.

• Overflow :-

Sometimes new data are to be

inserted into a data structure but there is no available space i.e. free storage list is empty. This situation is called as overflow (avail = null)

- Underflow :-

The term underflow refers to the situation where one wants to delete data from data structure i.e. empty (Start = null)

- * INSERTION :- at beginning of list :-

INSERTFIRST (INFO, LINK, START, AVAIL, ITEM)

1) IF AVAIL := NULL then write :-
OVERFLOW and Exit.

2) Set NEW := AVAIL and AVAIL :=
LINK[AVAIL]

3) Set INFO[NEW] := ITEM

4) Set LINK[NEW] := START

5) Set START := NEW

6) Exit

* Inserting after given node :-

INSLOC (INFO, LINK, LOC, AVAIL, ITEM)

1) IF AVAIL := NULL then write
:- Overflow and Exit

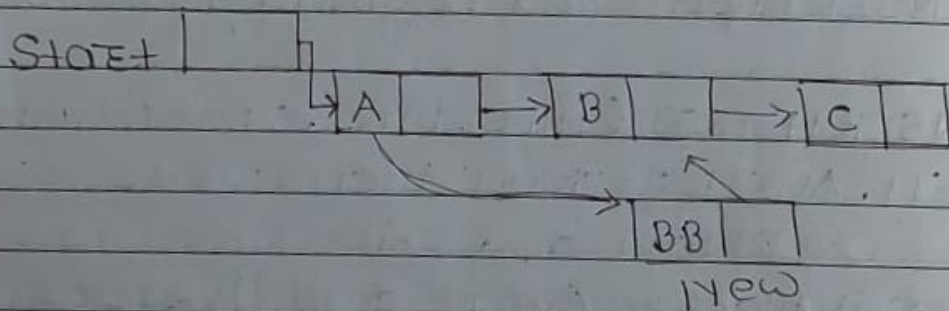
2) Set New := AVAIL and AVAIL :=
Link [AVAIL]

3) Set INFO [New] := ITEM

4) if LOC := NULL then Set LINK [New]
= START and Start := New

Else : Set Link [New] := LINK [LOC]
and Link [LOC] := New

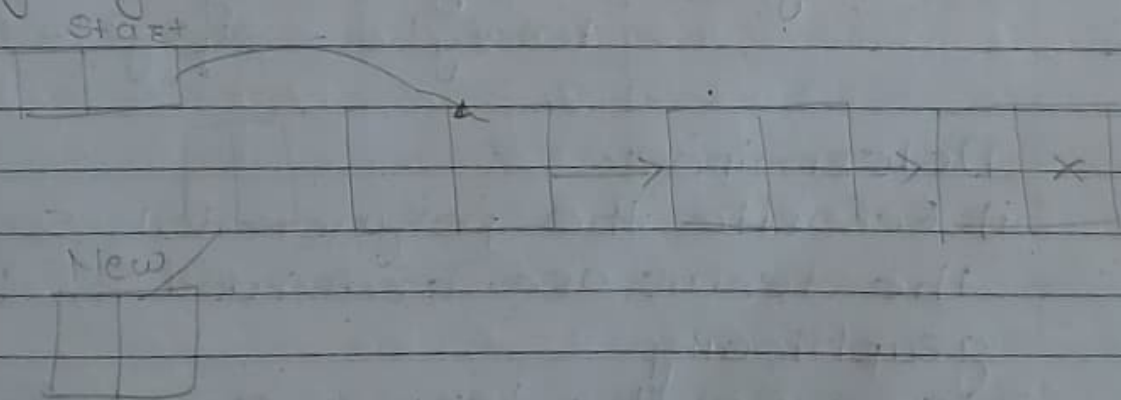
5) Exit



- Deletion

Deleting the node following the given node

Beginning Deletion



DEL (TNFO, Link, Start, Avail, Loc, LocP)

1) if LocP = Null then

Set Start := Link [Start]

else

Set link [LocP] := Link [Loc]

2) Set Link [Loc] = Avail and Avail := Loc

3) Exit

Q. Consider a polynomial function

$P(x, y, z)$ in variable x, y, z

$$P(x, y, z) = 8x^2y^2z - 6yz^8 + 3x^3yz + 2xy^7z - 5y^2y^3 - 4xy^7z^3$$

Determine :-

- Rewrite the polynomial so that the terms are ordered in Lexicographically.
- Suppose the terms are stored in the array COEF XEXP YEXP ZEXP with the head node first. Assign value to link so that linked list contain the ordered sequence of term.

i) Ans :-

$$\Rightarrow 3x^3yz - 5x^2y^3 + 8x^2y^2z - 4xy^7z^3 + 2xy^7z - 6x^0yz^8$$

ii) Ans: - Start COEF XEXP YEXP ZEXP Link

1			-1	-1	-1	2
2	3	3	1	1	3	3
3	-5	2	0	3	0	4
4	8	2	1	2	1	5
5	-4	1	3	7	3	6
6	2	1	1	7	1	7
7	6	0	1	8	8	-1
8						

Randomly saved

Q. $P(x, y, z) = 2xy^2z^3 + 3x^2yz^2 + 4xy^3z + 5x^2y^2 + 6y^2z + 7x^3z + 8xy^2z^5 + 9$

Sequentially saved

i) Rewrite polynomial so that terms are ordered lexicographically.

ii) Suppose terms are stored in array with circular head node first. Assign with header node first

Ans: - i) $7x^3z + 5x^2y^2 + 3x^2yz^2 + 4xy^3z + 8xy^2z^5 + 2xy^2z^3 + 6y^3z + 9$

Ans: -

ii) Start	COEF	XEXP	YEXP	ZEXP	Link
1		-1	-1	-1	6
2	8	1	2	5	4
3	5	2	2	0	5
4	2	1	2	3	8
5	3	2	1	2	7
6	7	3	0	1	3
7	4	1	3	1	2
8	6	0	3	1	9
9	9	0	0	0	1

Q. Let $P(x)$ denote the polynomial function $p(x) = 4x^9 + 2x^7 + 10x^2 - 2$

Draw a diagram to represent $P(x)$ by a header list with array structure

$P(x) =$ In Lexicographically order
 $4x^9 + 2x^7 - 15x^5 + 10x^2 - 2$

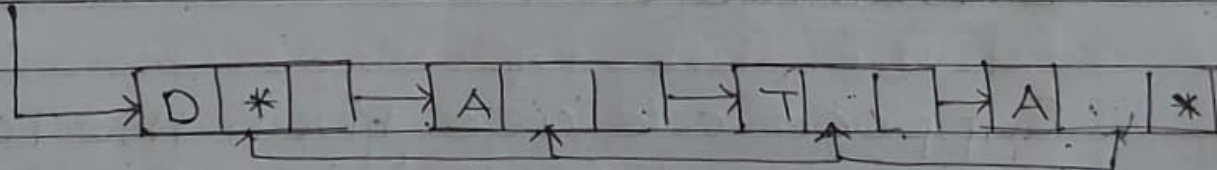
START		COEF	XEXP	LINK
↓	1	-1	-1	2
	2	4	9	3
	3	2	7	4
	4	-15	5	5
	5	10	2	6
	6	2	0	-1
	7			

Two way Linked List

A two way list is the linear collection of data element called as node. where each node is divided into three parts.

- i) An information filled INFO which contain data of n
- ii) A pointer filled FORW which contain the location of next node in the list.
- iii) A pointer filled Back which contain the location of preceding node in the list.

START



• Memory Representation of two way list.

Suppose five students in the class
A, B, C, D, E, F

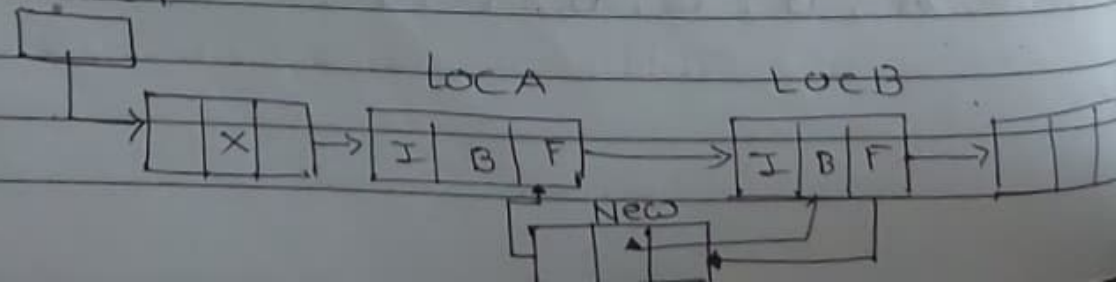
First		INFO	FORW	BACK
8	1	C	6	
	2		3	
	3		7	
LAST	4	E	-1	6
9	5	B	1	8
	6	D	4	5
Avail	7		9	
2	8	A	5	-1
	9		10	
	10		-1	

Note:- Traversing Searching operation of one way list are also applicable to two way list.

• INSERTION

Suppose we are given the Location Loc A and Loc B of adjacent nodes A and B in the list, and Suppose we want to insert a given atom or item of information between nodes A and B.

Start



INSTWL (INFO, BACK, FORW, ITEM, LOCA, LOCB, AVAIL, START)

1] IF AVAIL = NULL then Write :
OVERFLOW and Exit

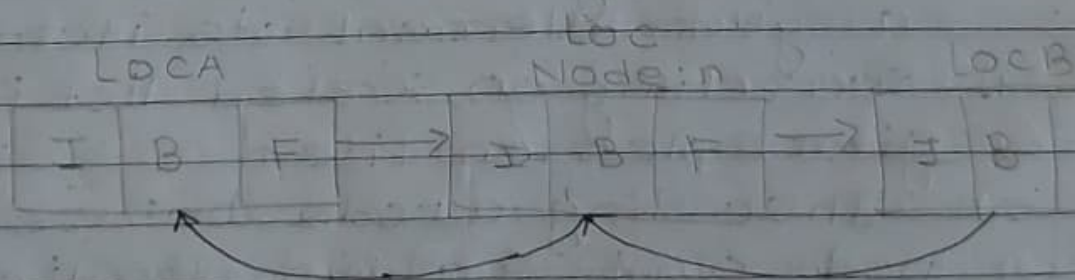
2] SET New = Avail ; Avail := FORW(Avail)
and INFO [New] = ITEM

3] ^{Set} FORW [LOCA] := New, FORW (New) := LOCB

4] BACK [LOCB] := New, BACK [New] := LOCA

DELETION :-

Suppose we are given the Location Loc of a node N in a list and we want to delete node N from the list.



DELTWL (INFO, BACK, FORW, AVAIL,
ITEM, LOC, LOCB, START)

1) Set FORW [LOC] := FEOW [LOC]
Back [LOCB] := Back [LOC]

2) Set FORW [LOC] := Avail and
Avail := LOC.

3) Exit

Q. Discuss the advantage of if any of two way list over one way list for each of the following operation

- 1) Traversing the list to process each node
- 2) Deleting the node whose location loc is given.
- 3) Searching an unsorted list for a given element or item.
- 4) Searching an sorted list for a given element item.
- 5) Inserting a node before the node with the given location loc
- 6) Inserting a node after the node with the given location loc

Ans :- 1) There is no advantage of two way list over one way list.

2) The location of preceding node is needed in one way list while two way list contain this information hence two way list is more efficient over one way list for deleting the node whose location loc is given.

3) There is no advantage of two way list over one way list.

4) There is no advantage of two way list over one way list.

5) As in ans :- 2

6) There is no advantage of two way list over one way list.