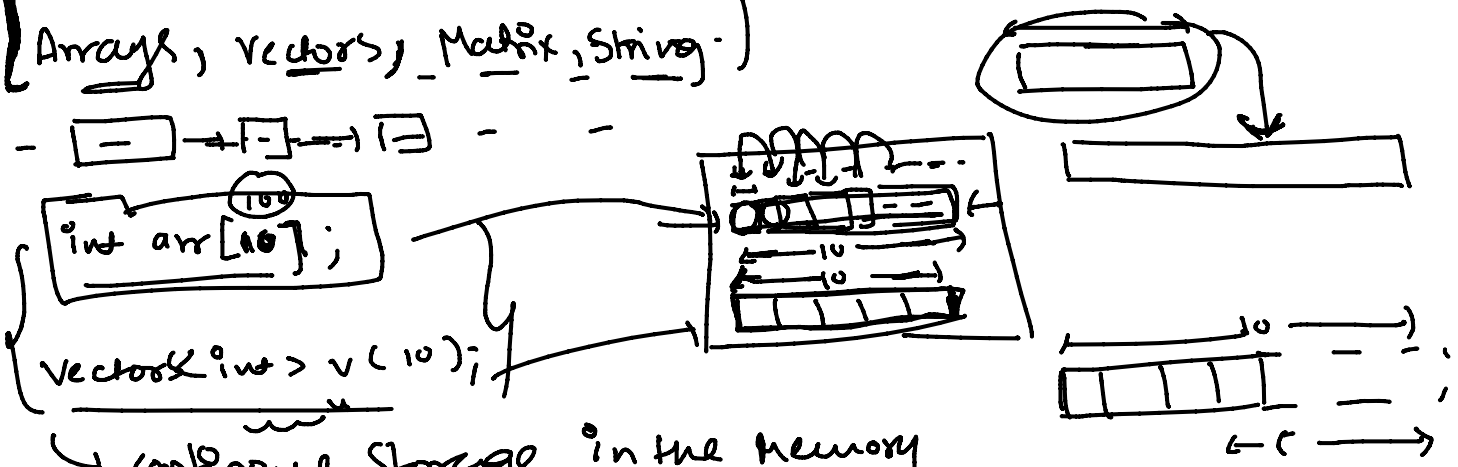
CLASS-49

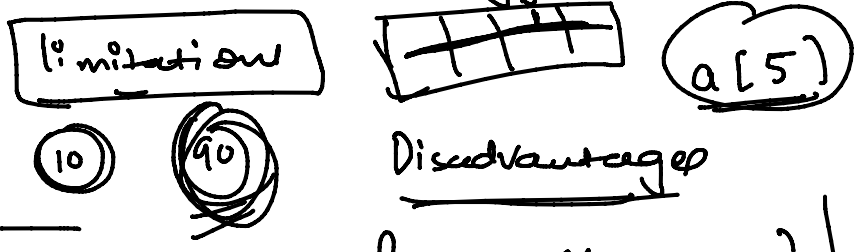
→ subProblem + work ⇒ code

linked List

[Arrays, Vectors, Matrix, String]



↳ continuous storage in the memory



Disadvantaged

→ wastage of memory
→ can't change memory runtime

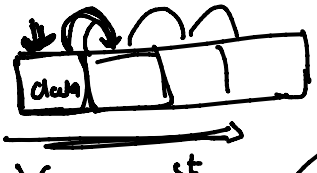
Advantaged

→ Random Access
→ Movement along the array/vector is easy

linkedlist

→ 1D storage (similar to Array/Vector)

Array →



→ continuous storage ✗

10 Blocks

10 Blocks

Address

Data | Add → Node

Store ?

Maie

Connected

6th Frd

6 Nodes

array | vector → Dynamically VS Static Memory Allocation

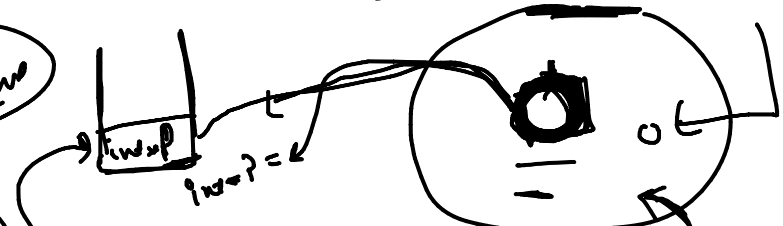
Compile time

Stack

.exe ← execute → Runtime

Heap

Compiletime



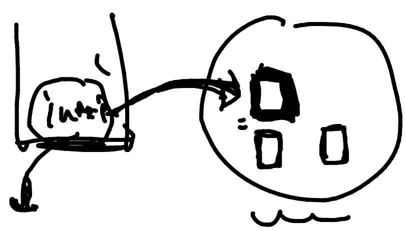
Dynamic

int

int * P = new int ;

→ Dynamic Memory Allocation → compress | extend

int function( )
{ int * P = new int ;
return P ;
}



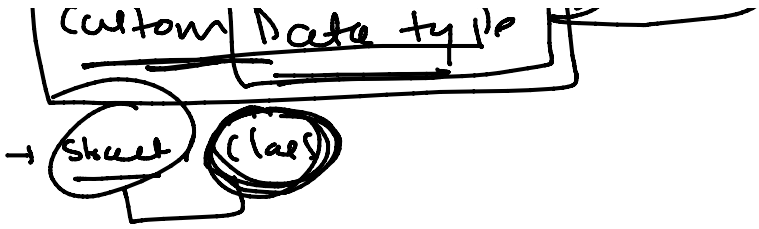
Memory leak

int → O

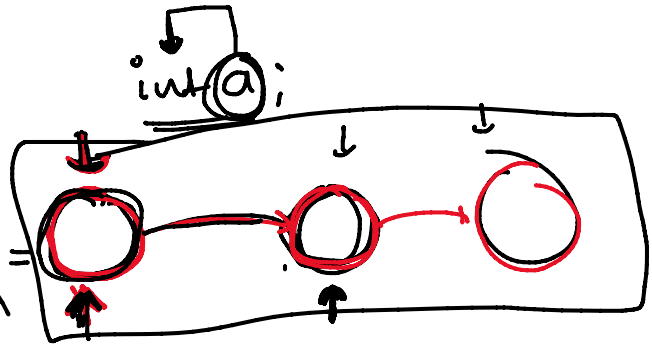
char → char

→ 1Block = Node → Data | Address

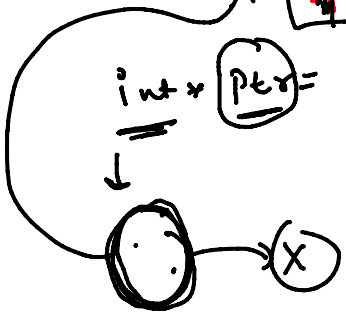
Custom Data type



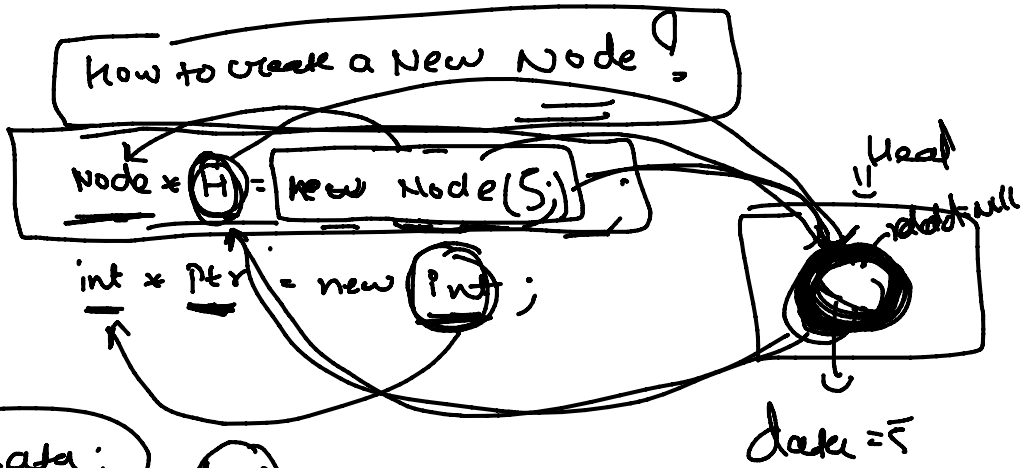
Create our own Data type




```
class Node
{ public:
    int data;
    Node * ptr;
Node (int d)
{ data = d;
  ptr = NULL
}
};
```

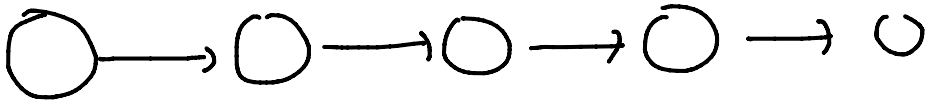


Char * ptr=



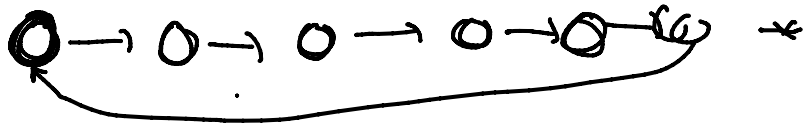
Types of Linked list

1) Singly linked list;



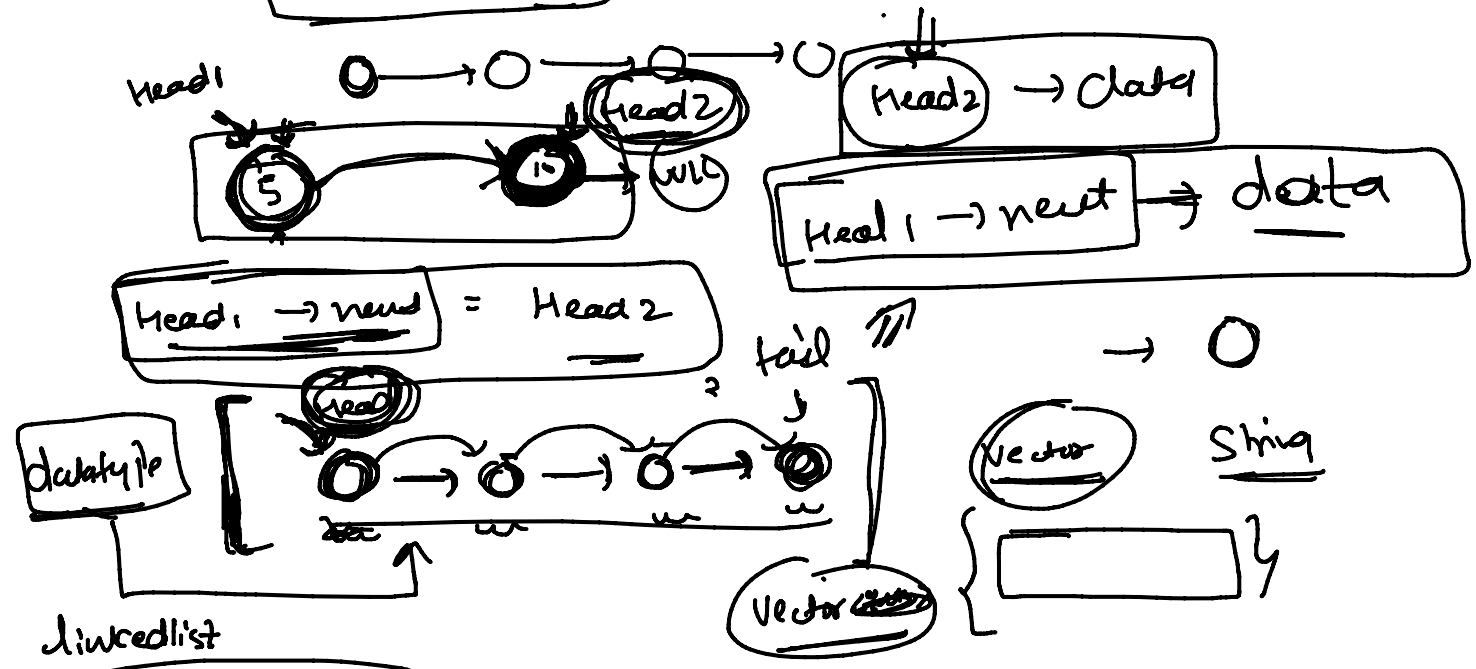
2) Doubly linked list

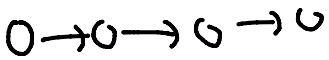
3) Circular Linked list



Node

linkedlist → Collection of Node





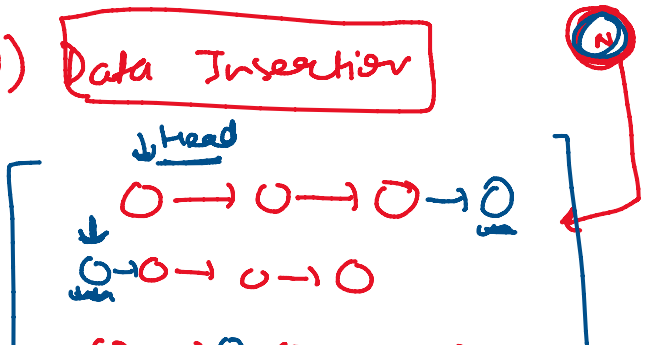
→ Operations on linked list

→ Data insertion

→ Access Data

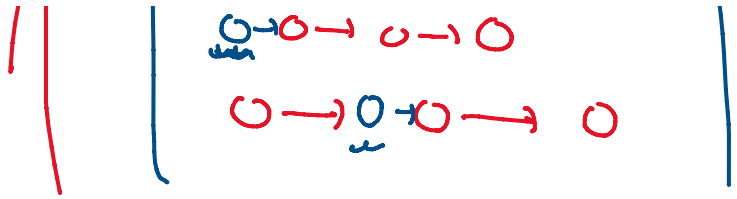
→ Data Deletion

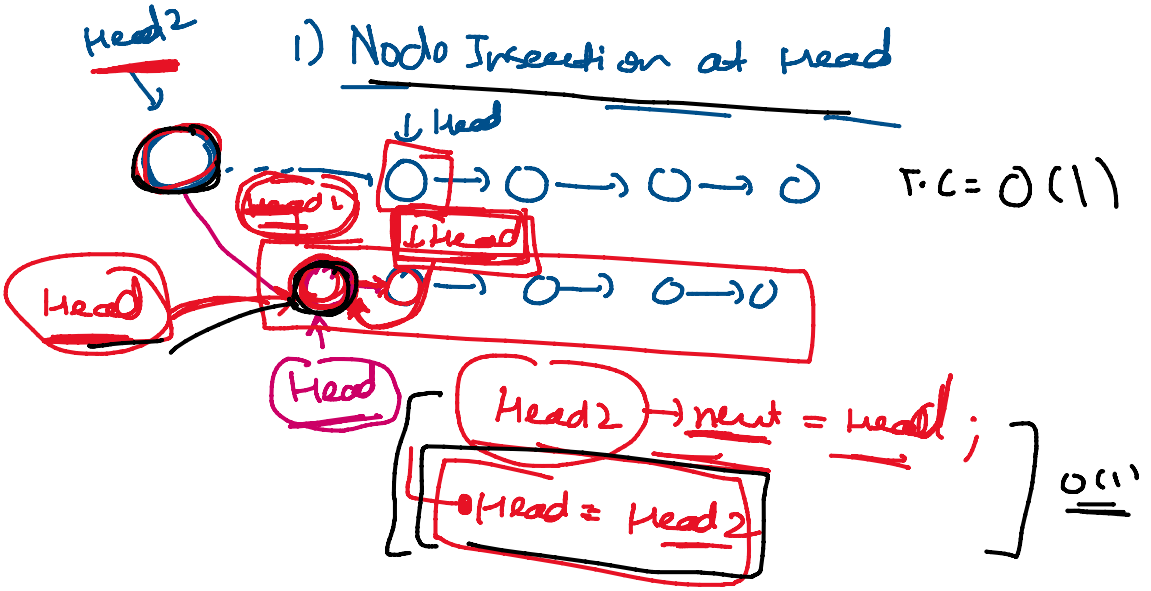
1) Data Insertion

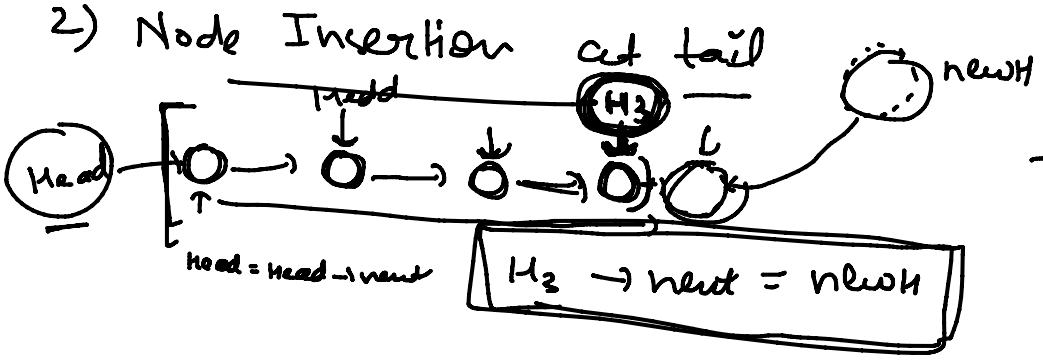


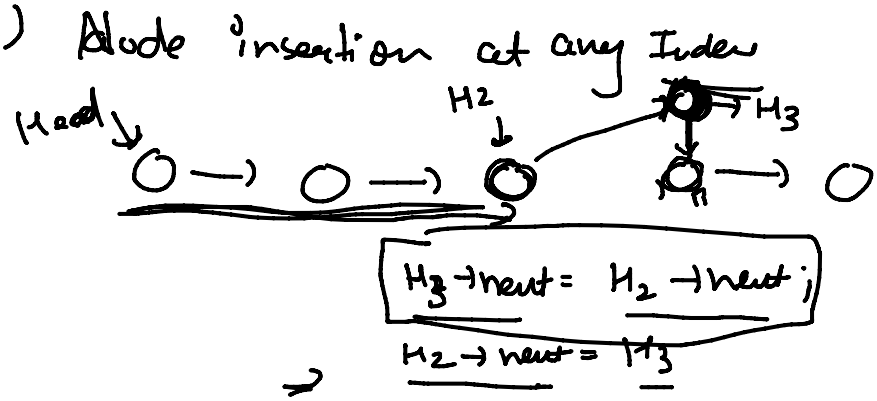
→ Data Deletion

→ Updation









Delete Nodes from linked list

1) Delete Node from Head

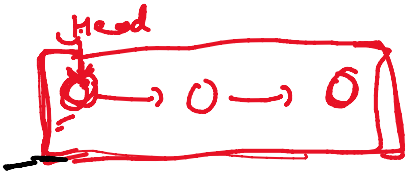
Head

2) Delete Node from Tail

head

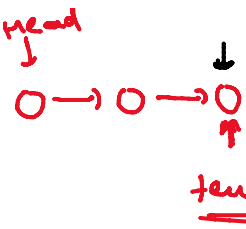
T.C = O(n)

// Delete Node from Head




```
Node * temp = Head;
Head = Head -> next;
delete temp
```

T.C: O(1)

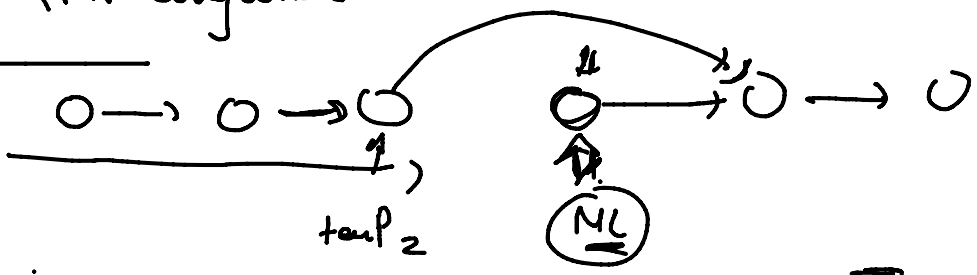


T.C = O(n)

```
temp -> next = NULL
delete temp;
```

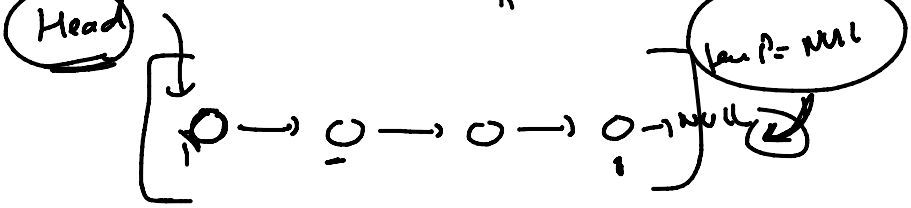
```
delete temp -> next
temp -> next = NULL;
```

3) Delete from anywhere



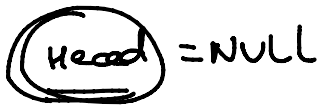
T.C = O(n)

```
temp2 -> next = temp1 -> next
delete temp1
```



* Printing the list

```
Node *temp = Head;
while (temp != NULL)
{ cout << temp -> data << " ";
  temp = temp -> next;
}
```


Head
=NULL