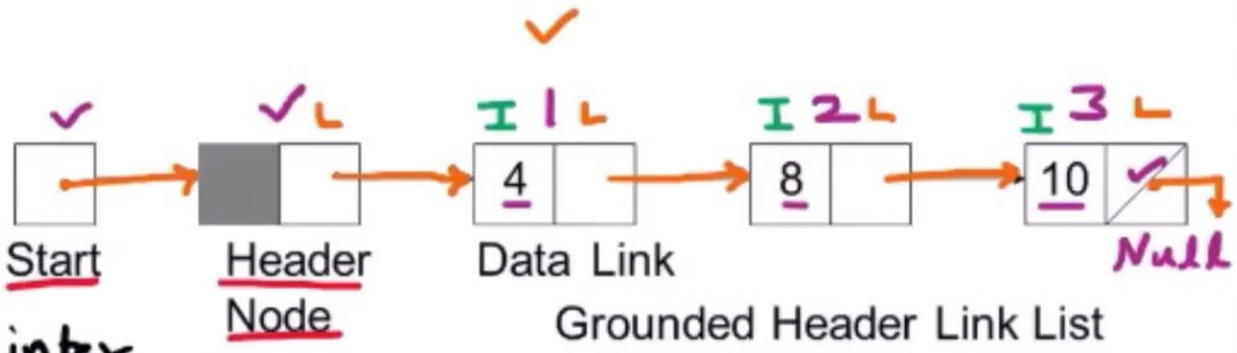
Header Linked List

- Contain special node called Header Node at begining of list

2 Types of Header Linked List

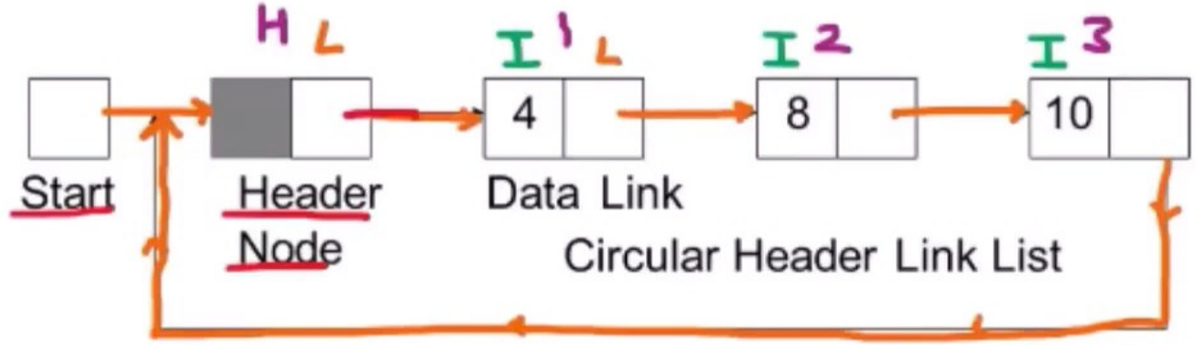
Grounded Header List

- Last node contain NULL pointer
- Term grounded indicate NULL pointer
- Start always point to Header Node
- LINK[START] = NULL means Grounded Header List is Empty



Circular Header List

- Last node point back to Header Node
- LINK[START] = START means Circular Header List is Empty
- Header list always be circular, unless stated
- Header node acting as sentinel indicating end of list

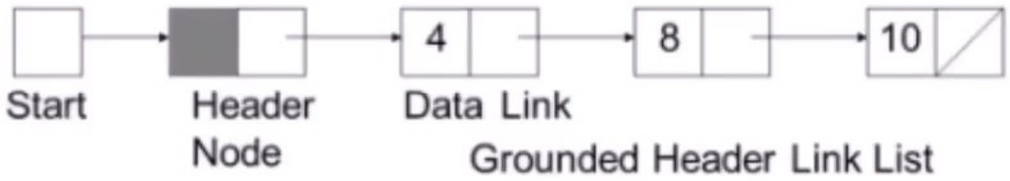


Header Linked List

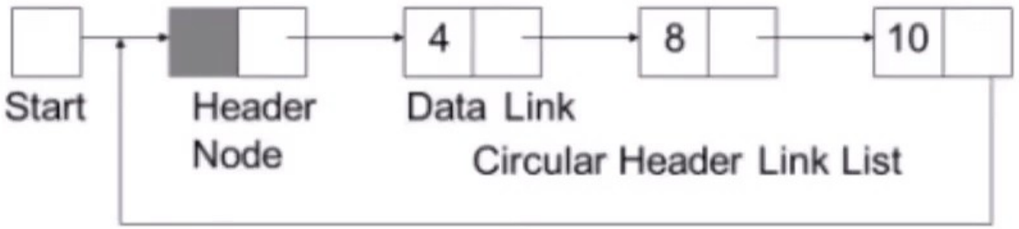
- Contain special node called Header Node at begining of list

2 Types of Header Linked List

Grounded Header List

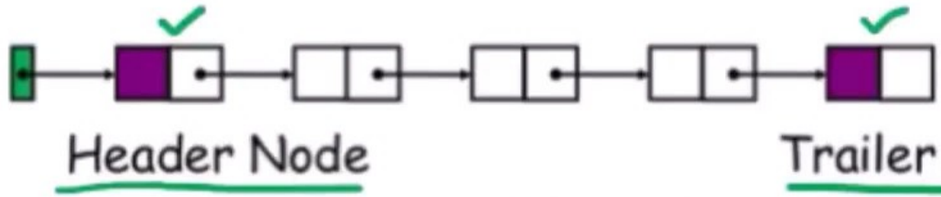


Circular Header List



Special Type of Header List

List with Header & Trailer Node



- Contain special Header Node at begining & Trailer Node at end of list

Important about Header List

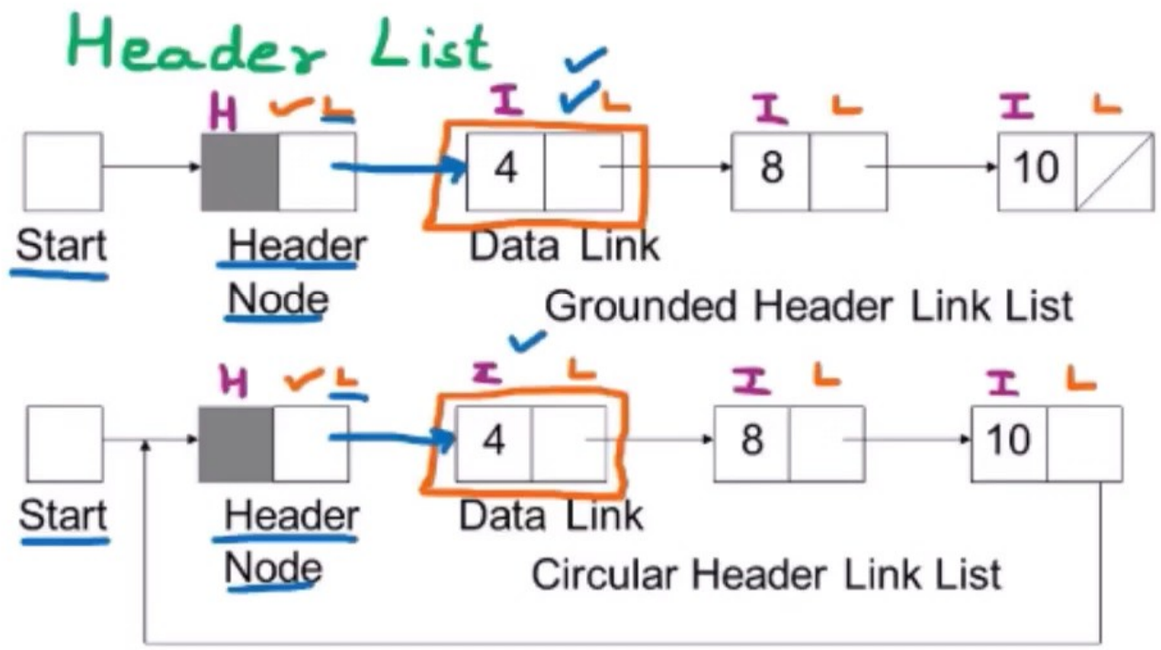
- Data may be maintained by Header List but AVAIL list will always be Ordinary List

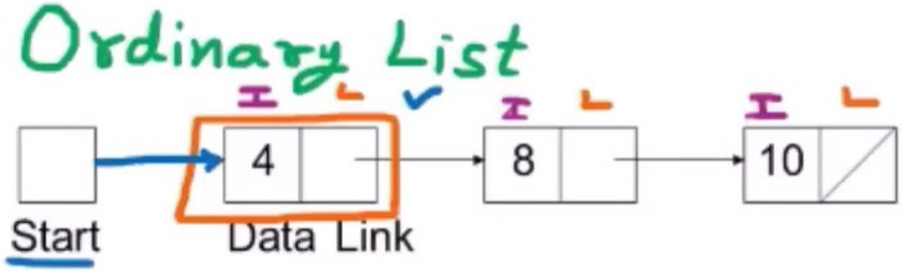
Header Linked List

- Contain special node called Header Node at begining of list

Header List vs Ordinary List

- first node is after Head Node, not like Ordinary List
- Location of First Node is LINK[START], not START like Ordinary List
- Circular Header List are easier to state & implement so more frequently used instead of Ordinary List
- Circular Header List not use NULL Pointer, so all pointer contain Valid address





Two Way List

- It is linear collection of data elements called nodes, where each node is divided into 3 Parts:

Information field (INFO)
- Contain Data

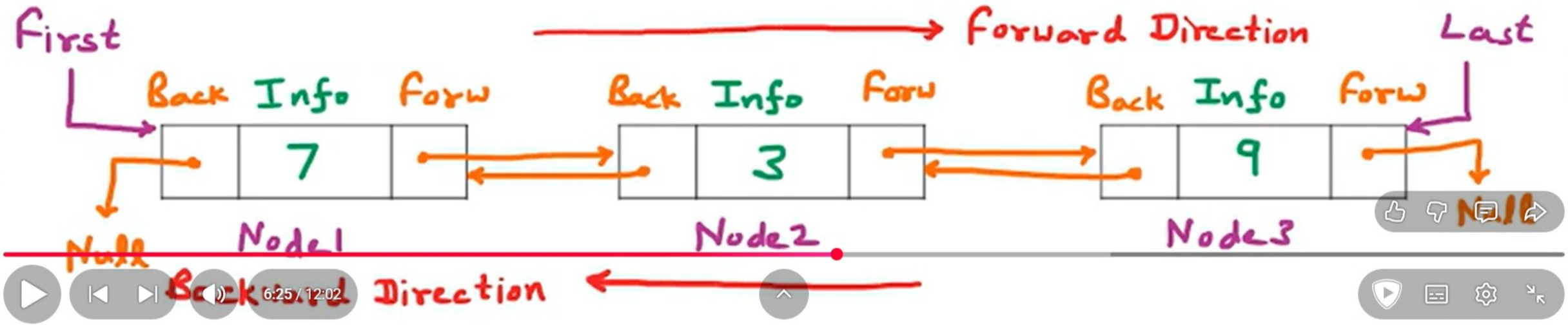
Forward Pointer (FORW)
- Contain Location of Next Node

Backward Pointer (BACK)
- Contain Location of Previous Node

List Require Two Pointers

First
- Points to First Node

Last
- Points to Last Node



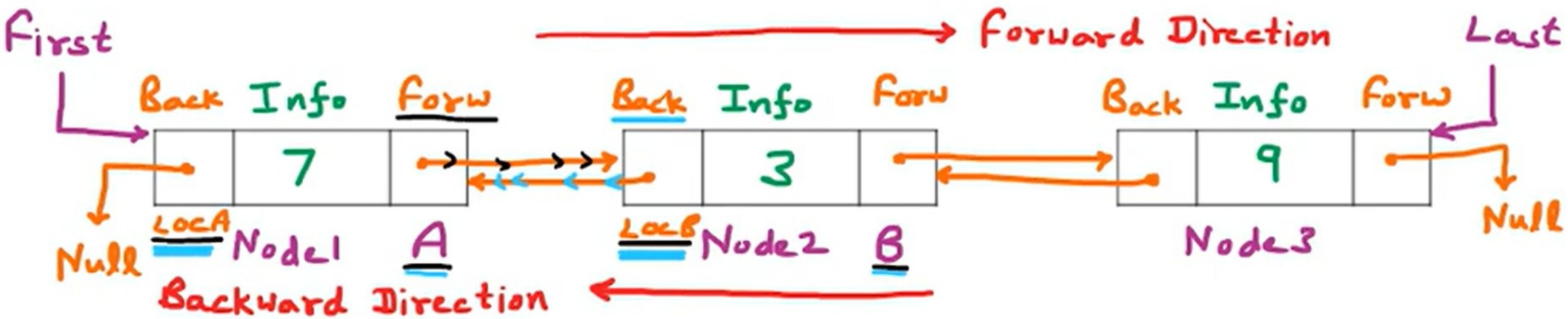
Two Way List

Move Forward & Backward

If Node B follows Node A and
Loc B & Loc A are their Locations

FORW [LOCA] = LOCB

BACK [LOCB] = LOCA



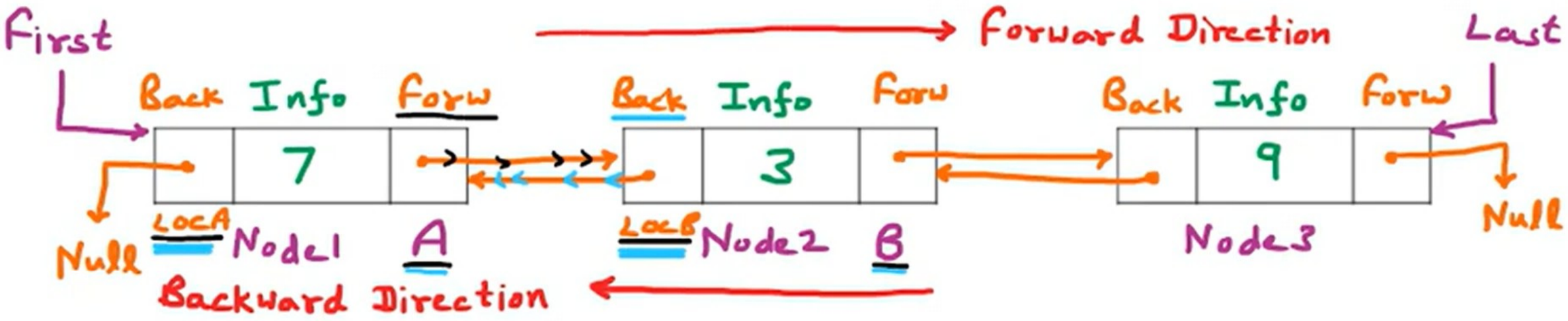
Two Way List

Disadvantages of Two Way List

- Storing data requires extra space for backward pointer & extra time to change the added pointer

Important about Two Way List

- Two way list require Two Pointers FORW & BACK, insted of one LINK pointer
- AVAIL list of available space still maintained as One Way List



Two Way Header List

- Two Way Circular Header List combine advantages of Two Way List & Circular Header List
- List is circular because two end nodes point back to Header Node
- Two Way Header List require only one pointer START, which point to Header Node
- Two pointer in Header Node points to two ends of list