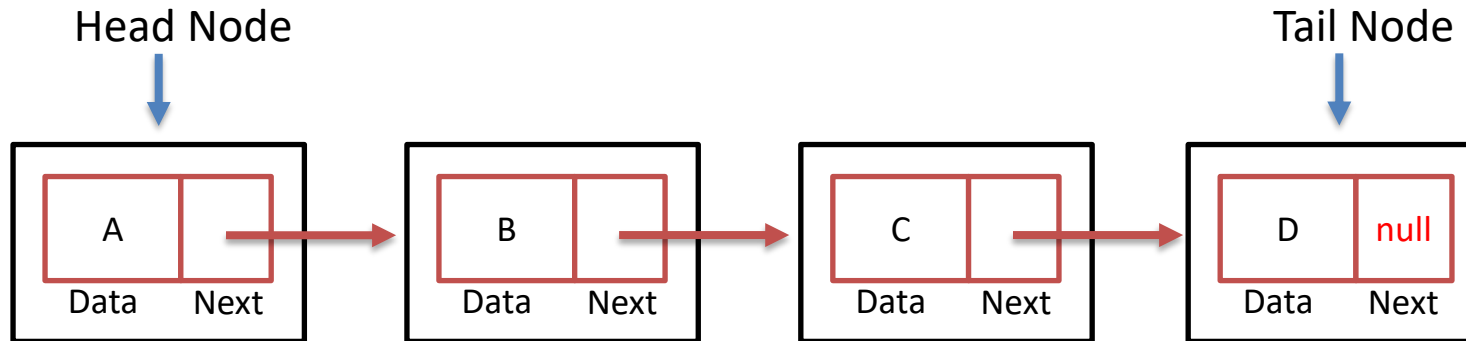


Singly Linked List



The **Head** of the LinkedList contains the reference/address of the first element in the LinkedList. The last node in the LinkedList is the **Tail**.

Singly-linked List stores the reference of only the next node in the LinkedList. The Last node does not refer any node that means last node contains “null” as next value.

Singly Linked List - Operations

Algo for Creating Linked List or Insertion at End of Linked List

Let's define a **Node** with two members **data** and **next**. Define two Node '**head**' and '**tail**' and set them to **NULL**.

1. Create a **newNode** with given value.
2. Check whether list is **Empty** (**head == NULL**).
 - a) If it is **Empty** then, set **head = newNode** and **tail = newNode**
 - b) Otherwise, set **tail -> next = newNode** and **tail = newNode**
3. Print list.

Singly Linked List - Operations

Algo for Traversing Linked List

Let **list** is a singly linked list that contains some nodes. Each **Node** contains two members **data** and **next**. Let two Node are '**head**' and '**tail**'.

1. Check, if(**head**==null)
 - a) Print, LinkedList is Empty
2. Otherwise,
 - a) Node **currentNode** = head
 - b) while (**currentNode**!=null)
 - i. Print **currentNode** >data
 - ii. Set **currentNode** = **currentNode** >next

Singly Linked List - Operations

Algo for Insertion at Beginning of Linked List

Let **list** is a LinkedList. '**head**' and '**tail**' are the nodes in the **list**.

1. Create a **list**.
2. Create a **newNode** with given value.
3. Check whether list is **Empty** (**head == NULL**).
 - a) If it is **Empty** then, set **head = newNode** and **tail = newNode**
 - b) Otherwise, set **newNode -> next = head** and **head = newNode**
4. Print list.

Singly Linked List - Operations



Algo for Insertion after Specific Node of Linked List

Let **list** is a LinkedList. '**head**' and '**tail**' are the nodes in the **list**. **locationNode** is the node after which we want to insert the **newNode**.

1. Create a **list**.
2. Create a **newNode** with given value.
3. set **newNode** -> next = **locationNode** -> next
4. set **locationNode** -> next = **newNode**
5. Print list.

Singly Linked List - Operations

Algo for Insertion after Specific Value of Node of Linked List

Let **list** is a LinkedList. '**head**' and '**tail**' are the nodes in the **list**. **LocationValue** is the node value after which we want to insert the **newNode**. **LocationNode** is the node of **LocationValue**.

1. Create a **list**.
2. Read **LocationValue**
3. Create a **newNode** with given value.
4. Traverse each node to find the location value.
 - a) set **currentNode** = **head**
 - b) set **locationNode** = null
 - c) While (**currentNode** != null)
 - i. value = currentNode -> data
 - ii. Check If (value == **LocationValue**) Then **locationNode** = **currentNode** and break loop
 - iii. currnetNode = currentttNode->next
5. Check if (**locationNode** == null) then Location Value does not exist.

Singly Linked List - Operations

continue.....

1. Otherwise,
 - a) set **newNode** -> next = **locationNode** -> next
 - b) set **locationNode** -> next = **newNode**
2. Print list.

Singly Linked List - Operations

Algo for Deletion from End of Linked List

Let's define a **Node** with two members **data** and **next**. Define two Node '**head**' and '**tail**' and set them to **NULL**.

1. Create a **newNode** with given value.
2. Check whether list is **Empty** (**head == NULL**).
 - a) If it is **Empty** then, set **head = newNode** and **tail = newNode**
 - b) Otherwise, set **tail -> next = newNode** and **tail = newNode**
3. Print list.

Singly Linked List - Operations



Algo for Deletion from Beginning of Linked List

Let's define a **Node** with two members **data** and **next**. Define two Node '**head**' and '**tail**' and set them to **NULL**.

1. Create a **newNode** with given value.
2. Check whether list is **Empty** (**head == NULL**).
 - a) If it is **Empty** then, set **head = newNode** and **tail = newNode**
 - b) Otherwise, set **tail -> next = newNode** and **tail = newNode**
3. Print list.

Singly Linked List - Operations

Algo for Deletion after Specific Value of Node of Linked List

Let's define a **Node** with two members **data** and **next**. Define two Node '**head**' and '**tail**' and set them to **NULL**.

1. Create a **newNode** with given value.
2. Check whether list is **Empty** (**head == NULL**).
 - a) If it is **Empty** then, set **head = newNode** and **tail = newNode**
 - b) Otherwise, set **tail -> next = newNode** and **tail = newNode**
3. Print list.