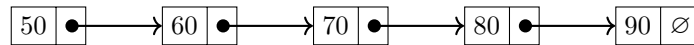


# Linked Lists Basics

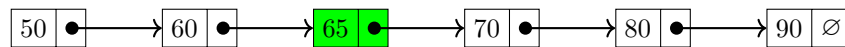
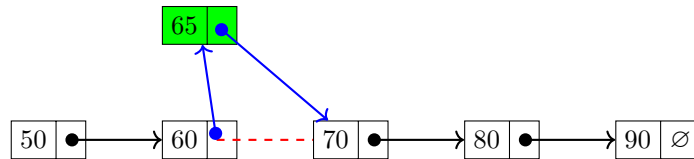
## 1 Singly Linked List:

### 1.1 Original Linked List:



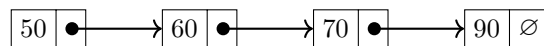
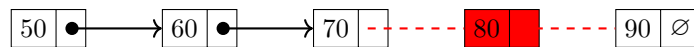
This linked list consists of five nodes. Each node has an integer value and a pointer that points to the next node in the linked list. The last node has a null pointer. It marks the end of the linked list.

### 1.2 Add node(65) to Linked List:



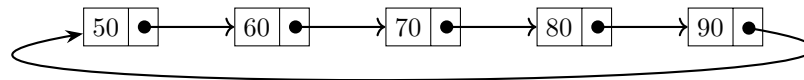
A new node is added to the linked list. It has an integer value of **65**. The node to be added is marked in green. The next pointer of the second node(**60**) now points to the new node(**65**). The next pointer of the new node(**65**) points to the third node(**70**) of the original linked list. The updated linked list has six nodes.

### 1.3 Delete node(80) from Linked List:



An existing node(80) is deleted from the linked list. The node to be deleted is marked in red. The next pointer of the third node(70) points to the last node(90). The fourth node(80) in the original linked list is removed. The updated linked list has four nodes.

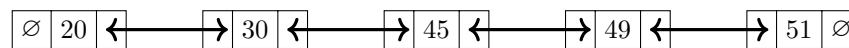
#### 1.4 Circular Linked List:



In a singly circular linked list, the next pointer of the last node(90) points to the first node(50).

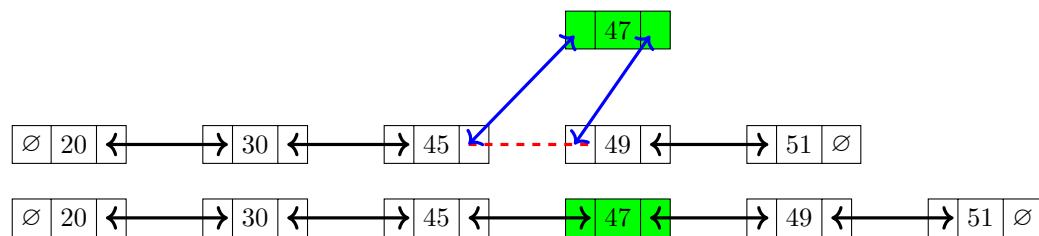
## 2 Doubly Linked List:

#### 2.1 Original Linked List:



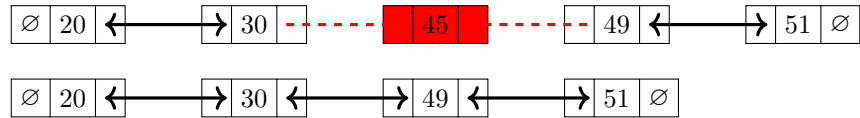
This linked list has five nodes. Each node has a previous pointer, an integer value and a next pointer. The previous pointer of the first node is Null. The next pointer of the last node(51) is null, and it marks the end of the doubly linked list.

#### 2.2 Add node(47) to Linked List:



A new node is added to the doubly linked list. The new node is marked in green. The previous pointer of the new node(47) points to the third node (aka previous node,45) of the original linked list. The next pointer of the new node(47) points to the fourth node(49) of the original linked list. Lastly, the previous pointer of the fourth node(49) of the original linked list points to the new node(47).

### 2.3 Delete node(45) from Linked List:



An existing node(45) is deleted from the linked list. The node to be deleted is marked in red. The next pointer of the second node(30) points to the fourth node(49) of the original linked list. The previous pointer of the fourth node(49) points to the second node(30) of the linked list (refer to 2.1). The updated linked list has four nodes.