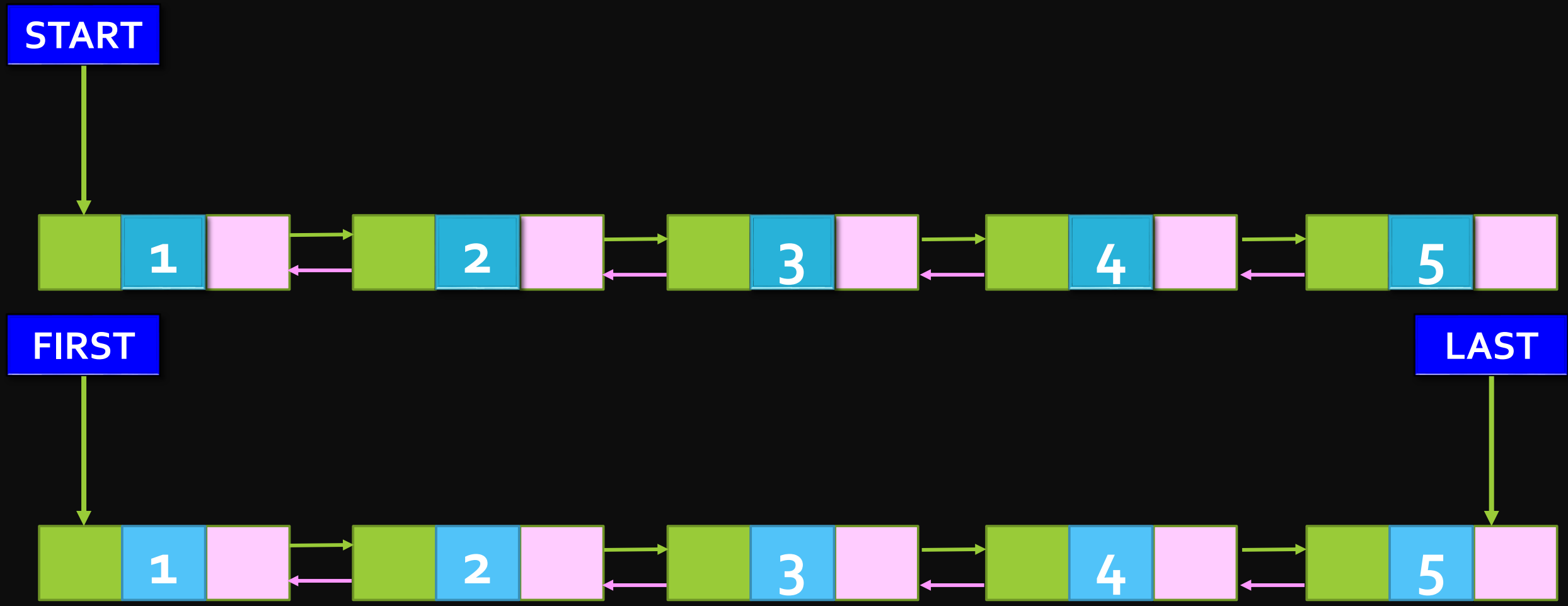
Data Structure (CS-303)

Doubly Linked List

DOUBLY LINKED LIST



DOUBLY LINKED LIST STRUCTURE OF NODE IN C

```
struct node
{
   struct node *prev;
   int data;
   struct node *next;
};
```

DOUBLY LINKED LIST

START

1

	DATA	PREV	NEXT
1	H	-1	3
2			
3	E	1	6
4			
5			
6	L	3	7
7	L	6	9
8			
9	O	7	-1

DOUBLY LINKED LIST INSERTION

Case 1: The new node is inserted at the beginning

Case 2: The new node is inserted at the end.

Case 3: The new node is inserted after a given node.

Case 4: The new node is inserted before a given node.

DOUBLY LINKED LIST INSERTION AT BEGINNING

Step 1: IF AVAIL = NULL

Write OVERFLOW

Go to Step 9

[END OF IF]

Step 2: SET NEW_NODE= AVAIL

Step 3: SET AVAIL = AVAIL → NEXT

Step 4: SET NEW_NODE → DATA = VAL

Step 5: SET NEW_NODE → PREV = NULL

Step 6: SET NEW_NODE → NEXT = START

Step 7: SET START → PREV = NEW_NODE

Step 8: SET START = NEW_NODE

Step 9: EXIT

DOUBLY LINKED LIST INSERT AT END

```
Step 1:   IF AVAIL = NULL
                  Write OVERFLOW
                  Go to Step 11
          [END OF IF]
Step 2:   SET NEW_NODE = AVAIL
Step 3:   SET AVAIL = AVAIL → NEXT
Step 4:   SET NEW_NODE → DATA = VAL
Step 5:   SET NEW_NODE → NEXT = NULL
Step 6:   SET PTR = START
Step 7:   Repeat Step 8 while PTR → NEXT != NULL
Step 8:           SET PTR = PTR → NEXT
          [END OF LOOP]
Step 9:   SET PTR →  NEXT = NEW_NODE
Step 10:  SET NEW_NODE → PREV = PTR
Step 11:  EXIT
```

DOUBLY LINKED LIST
INSERT AFTER GIVEN NODE

Step 1: IF AVAIL = NULL
Write OVERFLOW, and Go to Step 12
[END OF IF]
Step 2: SET NEW_NODE= AVAIL
Step 3: SET AVAIL = AVAIL → NEXT
Step 4: SET NEW_NODE → DATA = VAL
Step 5: SET PTR = START
Step 6: Repeat Step 7 while PTR → DATA != NUM
Step 7: SET PTR = PTR → NEXT
[END OF LOOP]
Step 8: SET NEW_NODE → NEXT = PTR → NEXT
Step 9 : SET NEW_NODE → PREV = PTR
Step 10: SET PTR → NEXT = NEW_NODE
Step 11: SET PTR → NEXT → PREV = NEW_NODE
Step 12: EXIT

DOUBLY LINKED LIST
INSERT BEFORE GIVEN NODE

Step 1: IF AVAIL = NULL
Write OVERFLOW, and Go to Step 12
[END OF IF]
Step 2: SET NEW_NODE= AVAIL
Step 3: SET AVAIL = AVAIL → NEXT
Step 4: SET NEW_NODE → DATA = VAL
Step 5: SET PTR = START
Step 6: Repeat Step 7 while PTR → DATA != NUM
Step 7: SET PTR = PTR → NEXT
[END OF LOOP]
Step 8: SET NEW_NODE → NEXT = PTR
Step 9 : SET NEW_NODE → PREV = PTR → PREV
Step 10: SET PTR → PREV = NEW_NODE
Step 11: SET PTR → PREV → NEXT = NEW_NODE
Step 12: EXIT

DOUBLY LINKED LIST DELETION

Case 1: The first node is deleted.

Case 2: The last node is deleted.

Case 3: The node after a given node is deleted.

Case 4: The node before a given node is deleted.

DOUBLY LINKED LIST
DELETE THE FIRST NODE

Step 1: If START = NULL

Write UNDERFLOW

Go to Step 6

[End of If]

Step 2: Set PTR = START

Step 3: Set START = START → NEXT

Step 4: Set START → PREV = NULL

Step 5: Free PTR

Step 6: EXIT

DOUBLY LINKED LIST
DELETE THE LAST NODE

Step 1: If START = NULL
Write UNDERFLOW
Go to Step 7
[End of If]
Step 2: Set PTR = START
Step 3: Repeat Step 4 while PTR → NEXT != NULL
Step 4: Set PTR = PTR → NEXT
[End of Loop]
Step 5: Set PTR → PREV → NEXT = NULL
Step 6: Free PTR
Step 7: EXIT

DOUBLY LINKED LIST
DELETE A NODE AFTER A GIVEN NODE

Step 1: If START = NULL
Write UNDERFLOW
Go to Step 9
[End of If]
Step 2: Set PTR = START
Step 3: Repeat Step 4 while PTR → DATA != NUM
Step 4: Set PTR = PTR → NEXT
[End of Loop]
Step 5: Set TEMP = PTR → NEXT
Step 6: Set PTR → NEXT = TEMP → NEXT
Step 7: Set TEMP → NEXT → PREV = PTR
Step 8: Free PTR
Step 9: EXIT

DOUBLY LINKED LIST
DELETE A NODE BEFORE A GIVEN NODE

Step 1: If START = NULL
Write UNDERFLOW
Go to Step 9
[End of If]
Step 2: Set PTR = START
Step 3: Repeat Step 4 while PTR → DATA != NUM
Step 4: Set PTR = PTR → NEXT
[End of Loop]
Step 5: Set TEMP = PTR → PREV
Step 6: Set TEMP → PREV → NEXT = PTR
Step 7: Set PTR → PREV = TEMP → PREV
Step 8: Free PTR
Step 9: EXIT