**DSA LAB**

**Lab Assignment number 11**

**Name:** Aamir Ansari **Batch:** A **Roll no:** 01

**Aim:** Implementation of Doubly linked list

**Theory:**

Doubly linked list?

A Doubly Linked List (DLL) contains an extra pointer, typically called previous pointer, together with next pointer and data which are there in singly linked list

Advantages over singly linked list

1) A DLL can be traversed in both forward and backward direction.

2) The delete operation in DLL is more efficient if pointer to the node to be deleted is given.

3) We can quickly insert a new node before a given node.

In singly linked list, to delete a node, pointer to the previous node is needed. To get this previous node, sometimes the list is traversed. In DLL, we can get the previous node using previous pointer.

**Algorithms:**

INSERT

**At the beginning**

Step 1: [INITIALIZE] newNode

Step 2: SET newNode->data = data

Step 3: IF start == NULL

SET newNode->next = NULL;

SET newNode->previous = NULL;

SET start = newNode;

Step 4: ELSE

SET newNode->next = start;

SET newNode->previous = NULL;

SET start->previous = newNode;

SET start = newNode;

[END IF]

Step 5: EXIT

**At the end**

Step 1: [INITIALIZE] newNode, ptr

Step 2: SET newNode->data = data

Step 3: IF start == NULL

SET newNode->next = NULL;

SET newNode->previous = NULL;

SET start = newNode;

Step 4: ELSE

SET ptr = end->next

Repeat while ptr->next != end

ptr=ptr-> next

[END LOOP]

SET ptr->next = newNode;

SET newNode->previous = ptr;

SET newNode->next = NULL;

[END IF]

Step 5: EXIT

**At a position:**

Step 1: [INITIALIZE] newNode, ptr

Step 2: SET newNode = start

Step 3: SET new->data = data

Step 4: IF start == NULL

PRINT “LIST EMPTY”

Goto Step 12

[END IF]

Step 4: SET count = 1

Step 5:Repaet step 6 to 8 while count!=position AND ptr->next!=end->next

Step 6: SET prePtr = ptr;

Step 7: SET ptr = ptr->next;

Step 8: count = count + 1

Step 9: IF count == 1

SET newNode->next = ptr;

SET newNode->previous = NULL;

SET ptr->previous = newNode;

SET start = newNode;

Step 10: ELSE IF ptr->next == end->next AND count < pos

SET ptr->next = newNode;

SET newNode->previous = ptr;

SET newNode->next = NULL;

Step 11:ELSE

SET newNode->next = ptr;

SET newNode->previous = ptr->previous;

SET ptr->previous->next = newNode;

SET ptr->previous = newNode;

[END IF]

Step 12:EXIT

**Before a given value:**

Step 1: [INITIALIZE] newNode, ptr

Step 2: SET newNode->data= data

Step 3: SET ptr=start

Step 4: IF end == NULL

PRINT “LIST IS EMPTY”

Goto Step 9

Step 5: Repeat step 6&7 while newNode->data != val

Step 6: SET ptr = ptr->next;

Step 7:IF ptr->previous == NULL

SET newNode->next = ptr;

SET newNode->previous = NULL;

SET ptr->previous = newNode;

SET start = newNode;

Step 8: ELSE

SET newNode->next = ptr;

SET newNode->previous = ptr->previous;

SET ptr->previous->next = newNode;

SET ptr->previous = newNode;

Step 9:EXIT

**After a given Value:**

Step 1: [INITIALIZE] newNode, ptr

Step 2: SET newNode->data= data

Step 3: SET ptr=start

Step 4: IF end == NULL

PRINT “LIST IS EMPTY”

Goto Step 10

Step 5: Repeat step 6&7 while ptr->data != val

Step 6: SET ptr = ptr->next;

Step 7: IF ptr->next == NULL

SET ptr->next = newNode;

SET newNode->previous = ptr;

SET newNode->next = NULL;

Step 8: ELSE

SET newNode->previous = ptr;

SET newNode->next = ptr->next;

SET ptr->next->previous = newNode;

SET ptr->next = newNode;

Step 9: EXIT

DELETE

**Value at the beginning**

Step 1: [INITIALIZE] ptr

Step 2: IF end == NULL

PRINT "LIST IS EMPTY"

Goto Step 6

[END IF]

Step 3: SET ptr = start

Step 4: IF ptr->next == NULL

SET start = NULL

Step 5: ELSE

SET ptr->next->previous = NULL;

SET start = ptr->next;

[END IF]

Step 5: free(ptr)

Step 6: EXIT

**At the end**

Step 1: [INITIALIZE] ptr

Step 2: IF start == NULL

PRINT "LIST IS EMPTY"

Goto Step 8

[END IF]

Step 3: Repeat Steps 4, 5 while ptr->next != NULL

Step 4: SET ptr = ptr->next;

[END LOOP]

Step 5: IF start->next == NULL

SET start = NULL

Step 6: ELSE

SET ptr->previous->next = NULL;

[END IF]

Step 7: free(ptr)

Step 8: EXIT

**Value at a Position**

Step 1: [INITIALIZE] ptr

Step 2: IF start == NULL

PRINT "LIST IS EMPTY"

Goto Step 12

[END IF]

Step 3: SET count = 1

Step 4: Repeat steps 5, 6 while count != pos AND ptr->next != NULL

Step 5: SET ptr = ptr->next;

Step 6: SET count = count + 1;

[END LOOP]

Step 7: IF POS > count OR pos <= 0

PRINT “NO NODE AVAILABLE”

Goto Step 12

[END IF]

Step 8: IF start->next == NULL

SET start = NULL

Step 9: ELSE IF count == 1

SET ptr->next->previous = NULL;

SET start = ptr->next;

Step 10: ELSE IF ptr->next == NULL

SET ptr->previous->next = NULL

Step 11: ELSE

SET ptr->previous->next = ptr->next;

SET ptr->next->previous = ptr->previous;

[END IF]

Step 12: EXIT

**Before a given value**

Step 1: [INITIALIZE] ptr

Step 2: IF start == NULL

PRINT "LIST IS EMPTY"

Goto Step 9

[END IF]

Step 3: IF start->data == val

PRINT “NO NODE BEFORE THIS”

Goto Step 9

[END IF]

Step 4: Repeat Step 5 while ptr->next->data != val

Step 5: SET ptr = ptr->next;

[END LOOP]

Step 6: IF ptr->previous == NULL THEN

SET ptr->next->previous = NULL;

SET start = ptr->next;

Step 7: ELSE

SET ptr->previous->next = ptr->next;

SET ptr->next->previous = ptr->previous;

[END IF]

Step 8: free(ptr)

Step 9: EXIT

**After a given value**

Step 1: [INITIALIZE] ptr

Step 2: IF start == NULL

PRINT "LIST IS EMPTY"

Goto Step 10

[END IF]

Step 3: Repeat Step 4 while ptr->data != val

Step 4: SET ptr = ptr->next;

[END LOOP]

Step 5: IF ptr->next == NULL THEN

PRINT “NO ELEMENT AFTER THIS”

[END IF]

Step 6: ptr = ptr->next

Step 7: IF ptr->next == NULL

SET ptr->previous->next = NULL;

Step 8: ELSE

SET ptr->previous->next = ptr->next;

SET ptr->next->previous = ptr->previous;

[END IF]

Step 9: free(ptr)

Step 10: EXIT

3.UPDATE

**Value at the beginning**

Step 1: IF start == NULL

PRINT "LIST IS EMPTY"

Goto Step 3

[END IF]

Step 2: SET start->data = toUpdate;

Step 3:EXIT

**At the end**

Step 1: IF start == NULL

PRINT "LIST IS EMPTY"

Goto Step 5

[END IF]

Step 2: Repeat Step 3 while ptr->data != NULL

Step 3: SET ptr = ptr->next;

[END LOOP]

Step 4: SET ptr->data = toUpdate;

Step 5:EXIT

**Value at a given Position**

Step 1: IF start == NULL

PRINT "LIST IS EMPTY"

Goto Step 9

[END IF]

Step 2: SET count = 1

Step 3: [INITIALIZE] ptr

Step 4: Repeat step 5, 6 while count != pos AND ptr->next!= NULL THEN

Step 5: SET ptr = ptr->next

Step 6: SET count = count + 1

[END LOOP]

Step 7: IF pos > count OR pos<=0 THEN

PRINT “NO NODE AT GIVEN POSITION”

Goto Step 9

[END IF]

Step 8: SET ptr->data = toUpdate

Step 9: EXIT

**Before a particular value**

Step 1: IF start == NULL

PRINT "LIST IS EMPTY"

Goto Step 7

[END IF]

Step 2: [INITIALIZE] ptr

Step 3: IF start->data == val THEN

PRINT “NO NODE BEFORE THIS”

Step 4: Repeat step 5 while ptr->next->data != val

Step 5: SET ptr = ptr->next

[END LOOP]

Step 6: SET ptr->data = toUpdate;

Step 7: EXIT

**After a particular value**

Step 1: IF end == NULL

PRINT "LIST IS EMPTY"

Goto Step 8

[END IF]

Step 2: [INITIALIZE] ptr

Step 3:Repeat step 4&5 while ptr->next->data != val

Step 4: SET ptr = ptr->next

[END LOOP]

Step 5: IF ptr->next == NULL THEN

PRINT “NO NODE AFTER THIS”

Step 6: SET ptr = ptr->next

Step 7: SET ptr->data = toUpdate;

Step 8: EXIT

4. COUNT NODES

Step 1: INITIALIZE count = 0, pr = START

Step 2: Repeat step 3&4 ptr->next != NULL

Step 3: SET count = count + 1

Step 4: SET ptr = ptr->next

Step 5: RETURN count

Step 6: EXIT

5. SEARCH

Step 1: SET PTR = START

Step 2: Repeat Step 3 while PTR != NULL

Step 3: IF VAL = PTR->DATA

PRINT ‘ELEMENT FOUND’

Go To Step 5

Step 4: ELSE

SET PTR = PTR->NEXT

[END OF IF]

Step 5: PRINT ‘ELEMENT NOT FOUND’

Step 6: EXIT

6. SORT

Step 1:[INITIALIZE] node traverse , min , temp

Step 2:Repeat step 3&4 while START->next

Step 3: SET min = START

Step 4: SET traverse = START->next

Step 5:Repeat step 6&7 while traverse is true

Step 6: IF min->data > traverse->data

SET min = traverse

Step 7: SET traverse = traverse->next

Step 8: SET temp = START->data

Step 9: SET START->data = min->data

Step 10: SET min->data = temp

Step 11: SET START = START->next

Step 12:EXIT

7. REVERSE

Step 1:[INITIALIZE] prev, ptr, next

Step 2:SET prev=NULL

Step 3:SET ptr=START

Step 4: Repeat step 5 to 8 while ptr!=NULL

Step 5: SET next=ptr->next

Step 6: SET ptr->next=prev

Step 7: SET prev=ptr

Step 8: SET ptr=next

Step 9:SET START=prev

8. DISPLAY

Step 1: [INITIALIZE] ptr

Step 2: IF start == NULL

PRINT "LIST IS EMPTY"

Goto Step 7

[END IF]

Step 3: Repeat Step 4, 5 while ptr->data != val

Step 4: SET ptr = ptr->next;

Step 5: PRINT ptr->data

[END LOOP]

Step 6: PRINT ptr->data

Step 7: EXIT