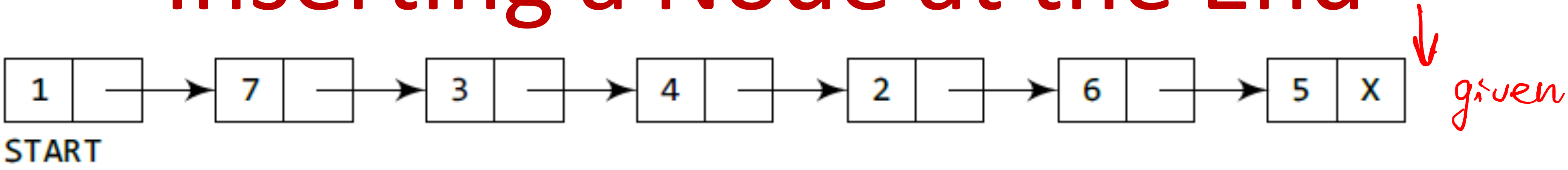
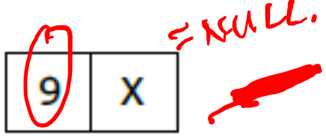
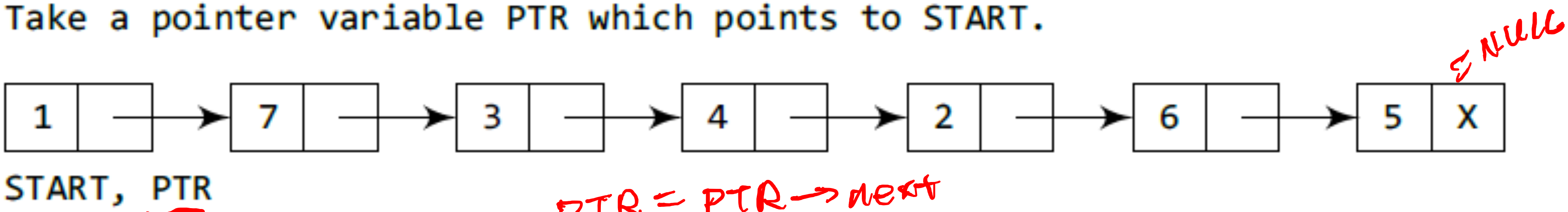
Inserting a Node at the End



Allocate memory for the new node and initialize its DATA part to 9 and NEXT part to NULL.

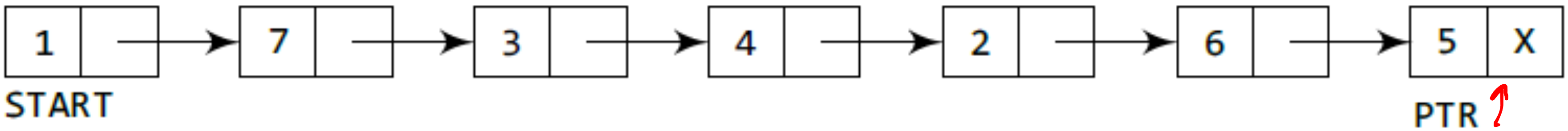


Take a pointer variable PTR which points to START.

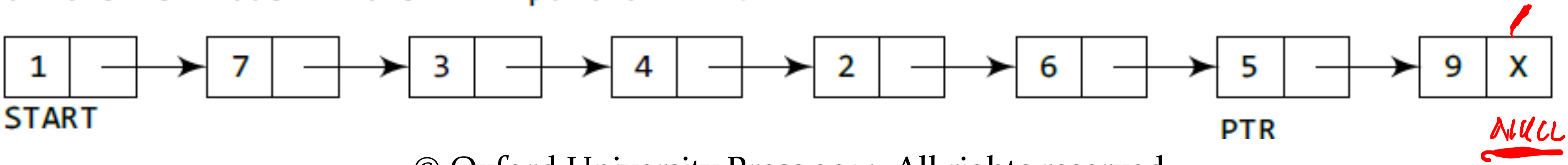


PTR = PTR→next

Move PTR so that it points to the last node of the list.



Add the new node after the node pointed by PTR. This is done by storing the address of the new node in the NEXT part of PTR.



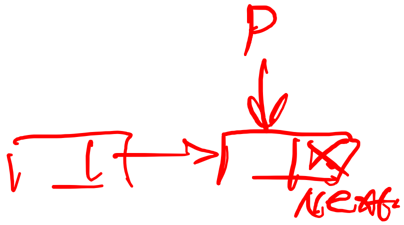
Step 1 : //check overflow & Create a NewNode → O(1)

Step 2 :

```
//empty
if (START == NULL) {            → O(1).
    START = newNode;
    return START;
}
```

Step 3 : //else traverse to the last node.

```
Node *p = START;
while (p → next != NULL){       → O(n).
      p = p → next;
}

// p → next == NULL
```



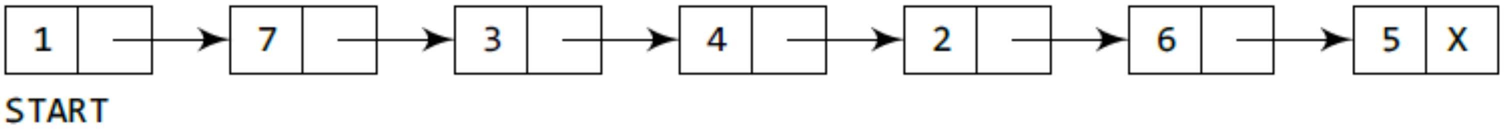
Step 4 : // insert, change tail node's next pointer

```
p → next = newNode;             → O(1)

return START;
```

Time complexity O(n).

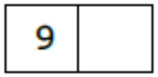
Inserting a Node at the End

```
ALGORITHM TO INSERT A NEW NODE AT THE END OF THE LINKED LIST

Step 1: IF AVAIL = NULL, then
                  Write OVERFLOW
                  Go to Step 10
         [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: EXIT
```

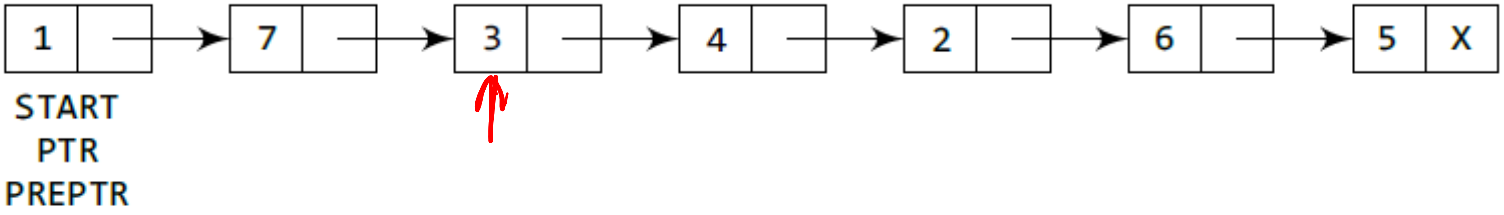


Allocate memory for the new node and initialize its DATA part to 9.

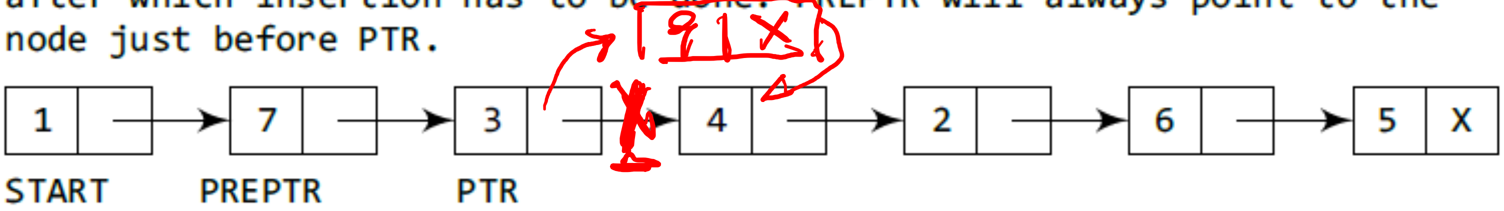
Inserting a Node after Node that has Value NUM (3)

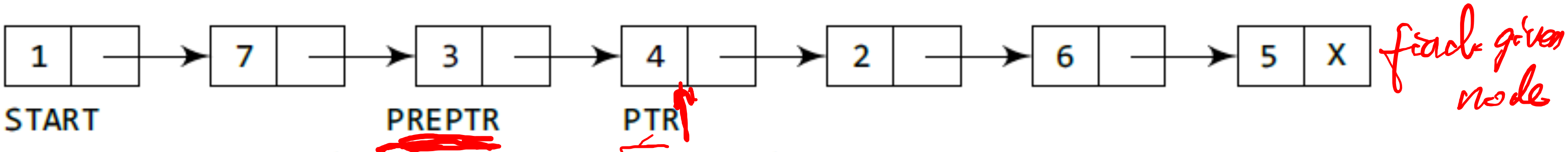


Take two pointer variables PTR and PREPTR and initialize them with START so that START, PTR, and PREPTR point to the first node of the list.

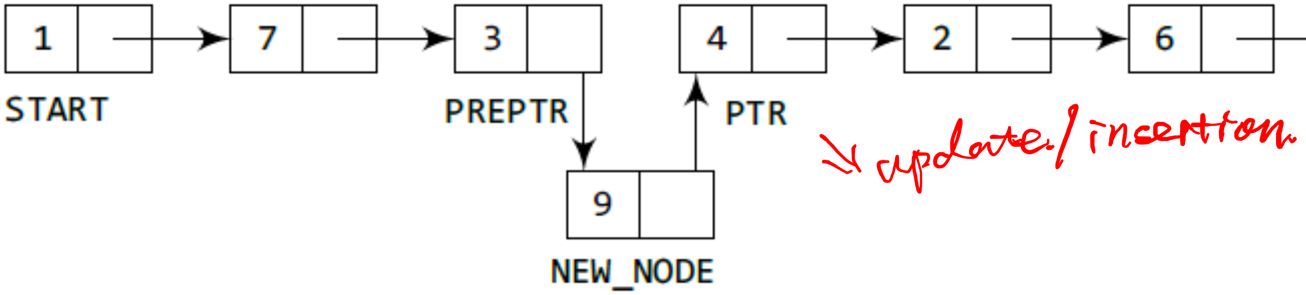


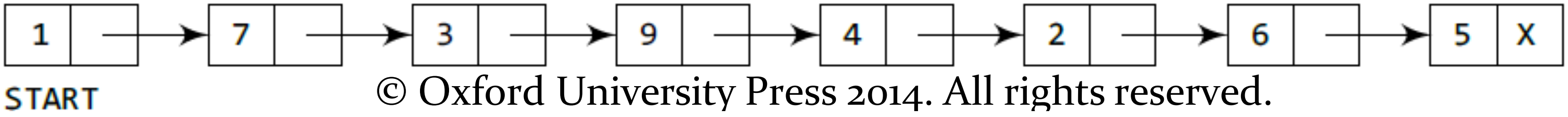
Move PTR and PREPTR until the DATA part of PREPTR = value of the node after which insertion has to be done. PREPTR will always point to the node just before PTR.





Add the new node in between the nodes pointed by PREPTR and PTR.





Insert after a given node (i.e 3).

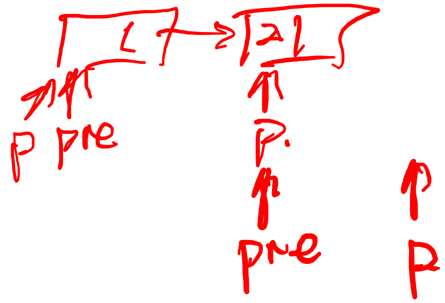
step1: // search the given node

```
Node* p = START;
Node *pre = START;
while (pre->data != 3 && p != NULL),
{
        pre = p;
        p = p->next;
              NULL.
}.
```

If (START == NULL).

pre != NULL && p != NULL).

O(n)



Time Complexity:
O(n).

```
if (pre->data == 3) // found
{  // insert
     pre->next = newNode;               -> O(1).
     newNode->next = p; -> connect the second part.
}    return START;
            p = NULL
else   // pre->data != 3 or No ~~data~~ give node found,
{
     printf("Not found given node"); -> O(1)
}    return;
```

Inserting a Node after Node that has Value NUM

```
ALGORITHM TO INSERT A NEW NODE AFTER A NODE THAT HAS VALUE NUM

Step 1: IF AVAIL = NULL, then
                 Write OVERFLOW
                 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: SET PREPTR = PTR
                                                  && PTR != NULL
Step 7: Repeat Steps 8 and 9 while PREPTR->DATA != NUM
Step 8:          SET PREPTR = PTR
Step 9:          SET PTR = PTR->NEXT
         [END OF LOOP]
Step 10: SET PREPTR->NEXT = New_Node
Step 11: SET New_Node->NEXT = PTR
Step 12: EXIT
```