

Lecture 9

Insertion Operation of Linked List

Prepared By:

Afsana Begum
Lecturer,
Software Engineering Department,
Daffodil International University.

Contents

- Insertion methods at any position
- Algorithm of Insertion

Inserting a new node

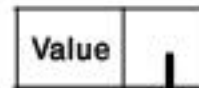
- Possible cases of Insert Node
 1. Insert into an empty list
 2. Insert in front
 3. Insert at back
 4. Insert in middle
- But, in fact, only need to handle two cases:
 1. Insert as the first node (Case 1 and Case 2)
 2. Insert in the middle or at the end of the list (Case 3 and Case 4)

Inserting at Beginning

Head



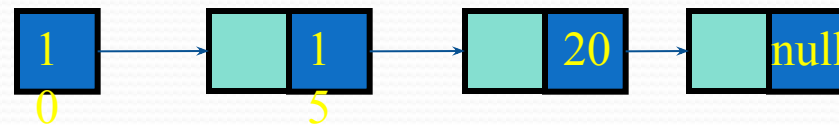
newNode 1



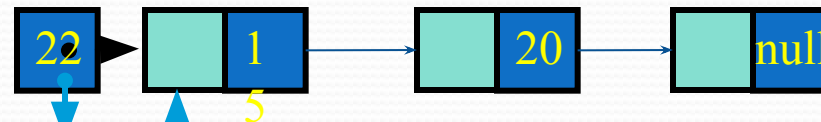
NULL

NULL

Start



START



NEW()

Inserting at Beginning

Step 1. OVERFLOW, IF(AVAIL = NULL)
write : OVERFLOW and EXIT.

Step 2. Create a new node NODE=AVAIL
and assign the address to any variable say AVAIL_PTR .

Step 3. ASSIGN NODE[DATA] = ITEM

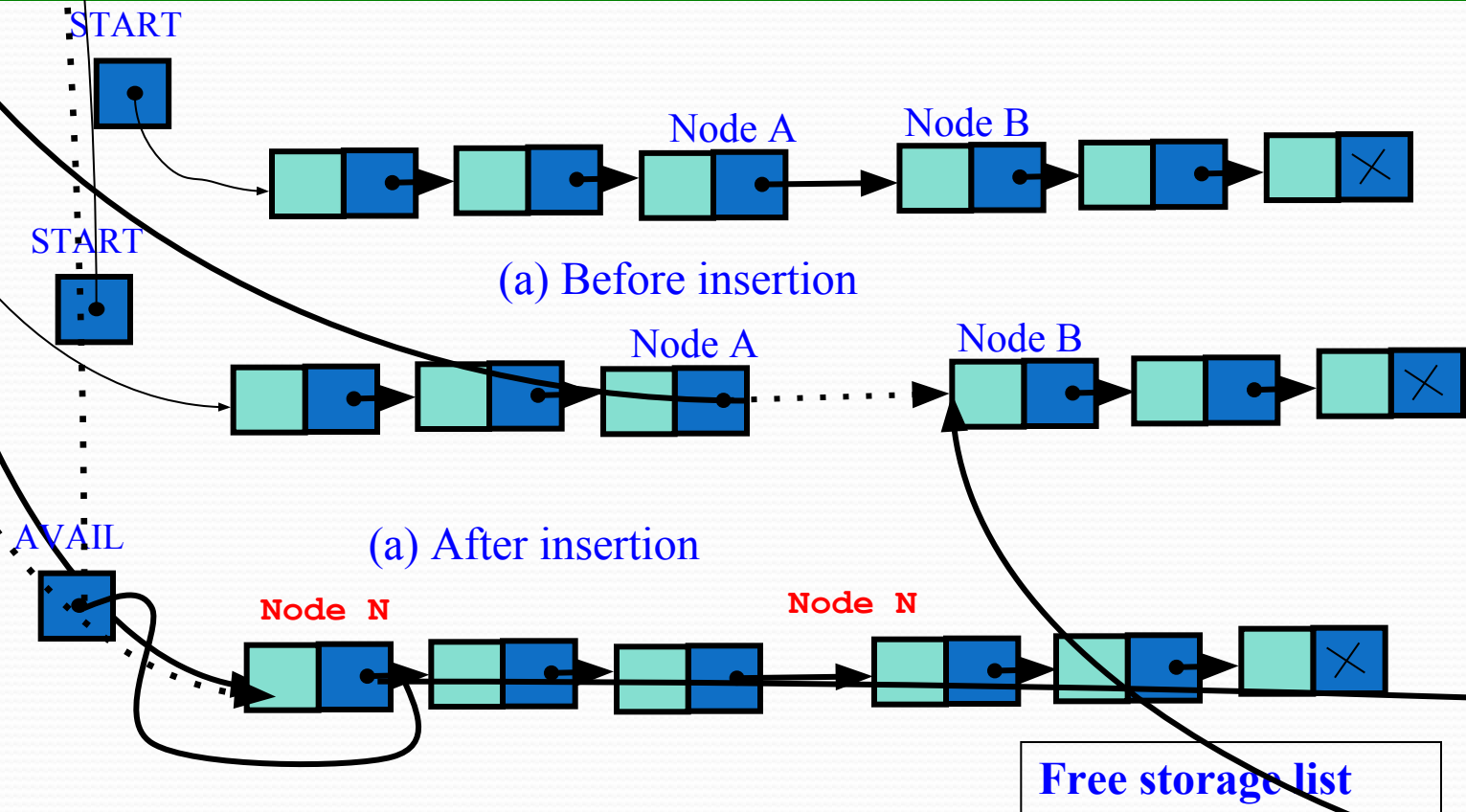
Step 4. IF(START = NULL)
ASSIGN NODE[LINK] = NULL
ELSE
ASSIGN NODE[LINK] = START

Step 5. ASSIGN START = AVAIL_PTR

Step 6. EXIT

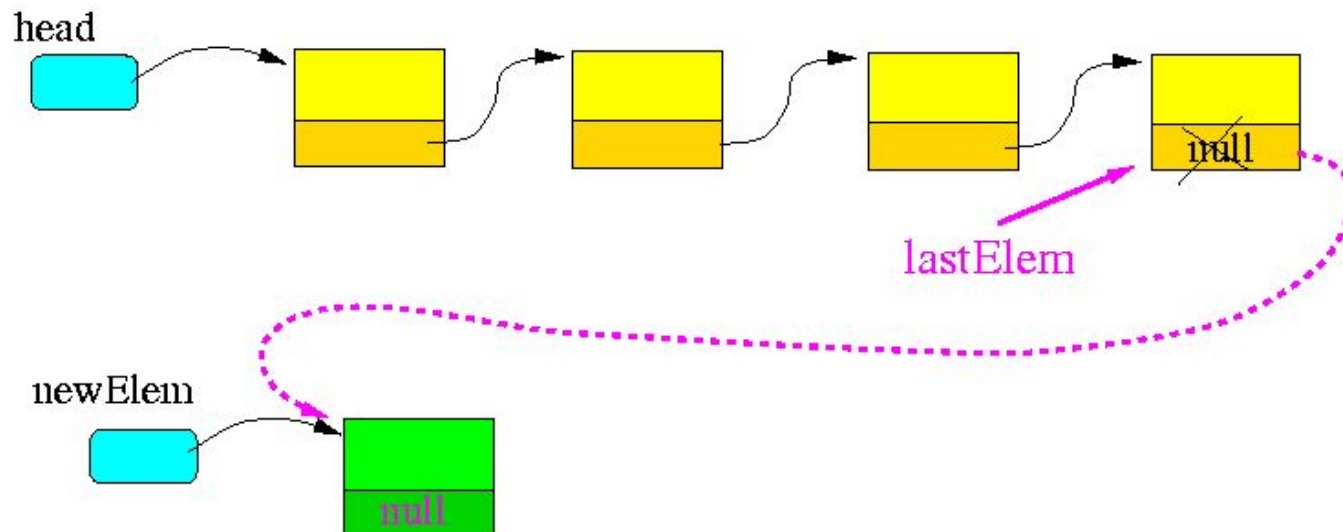
Insertion in the Middle

- Node N is to be inserted in to the list between nodes A and B
- Three pointer fields are changed as follows:
 1. The next pointer field of node A now points to the new node N, to which AVAIL previously pointed.
 2. AVAIL now point to the second node in the free pool, to which node N previously pointed.
 3. The next pointer field of node N now points to node B, to which node A previously pointed.



Insertion in the last position

What you need to do to insert at tail:



Inserting at middle or last

INSLASTLOC(INFO, LINK, START, AVAIL, LOC, ITEM)

1. [OVERFLOW?] If AVAIL=NULL, then print OVERFLOW and exit
2. Set NODE= AVAIL
3. Set NODE[DATA]= ITEM
4. IF PREVIOUS_NODE[LINK]=NULL
Set NODE[LINK]= NULL and PREVIOUS_NODE[LINK]=AVAIL_PTR

ELSE
Set NODE[LINK]= PREVIOUS_NODE[LINK] and PREVIOUS_NODE[LINK]=AVAIL_PTR
5. Exit.



Question?