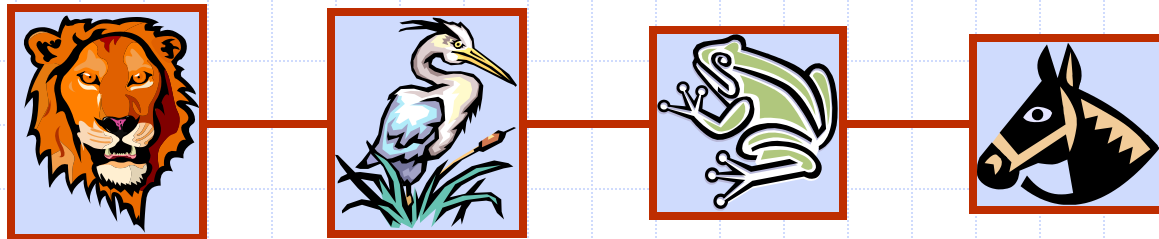


Presentation for use with the textbook **Data Structures and Algorithms in Java, 6<sup>th</sup> edition**, by M. T. Goodrich, R. Tamassia, and M. H. Goldwasser, Wiley, 2014

# Singly Linked Lists



# Reading

**M. T. Goodrich, R. Tamassia and M. H. Goldwasser,**  
*Data Structures and Algorithms in Java*, 6th Edition,  
2014.

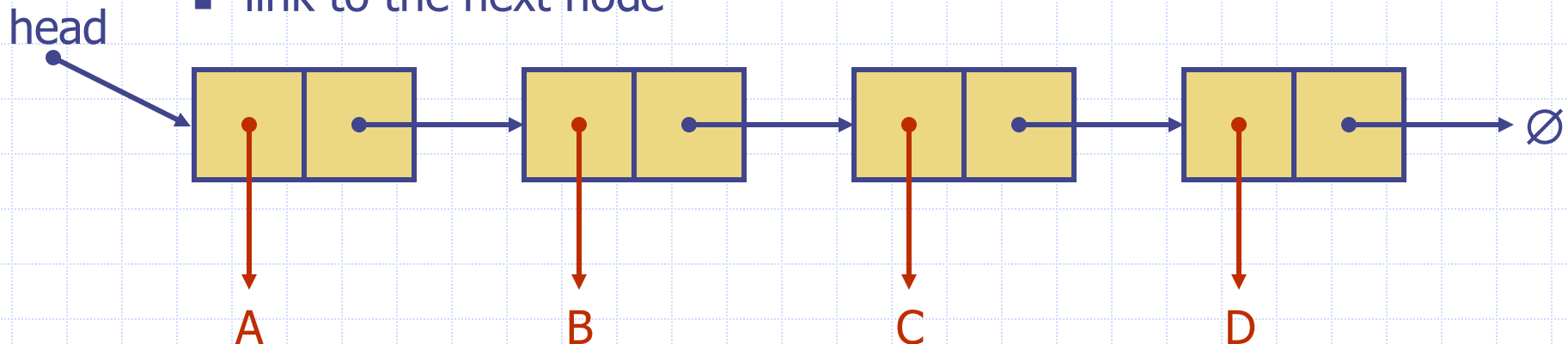
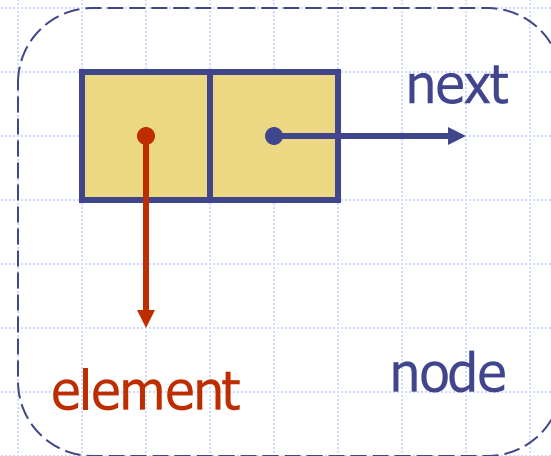
- Chapter 3. Arrays and Linked Lists

# Singly Linked List

◆ A singly linked list is a **concrete data structure** consisting of a sequence of nodes, starting from a head pointer

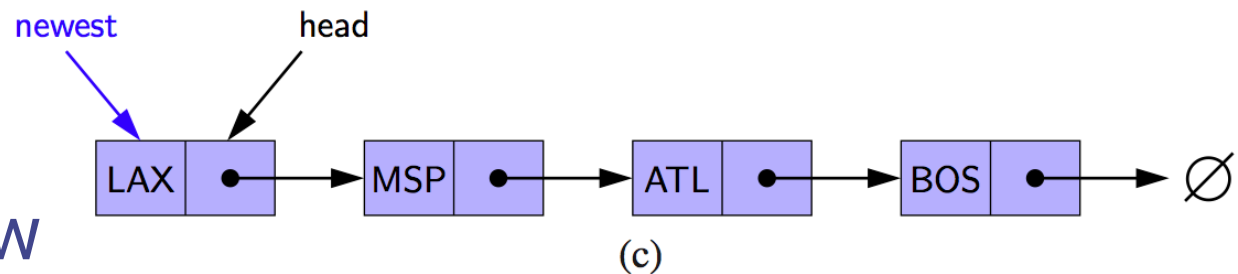
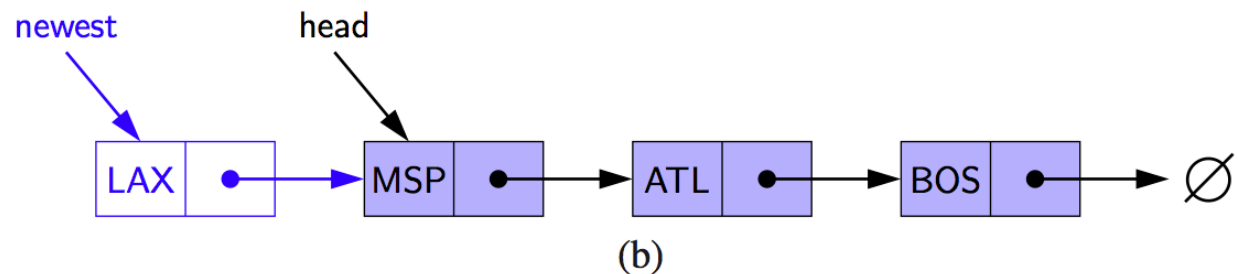
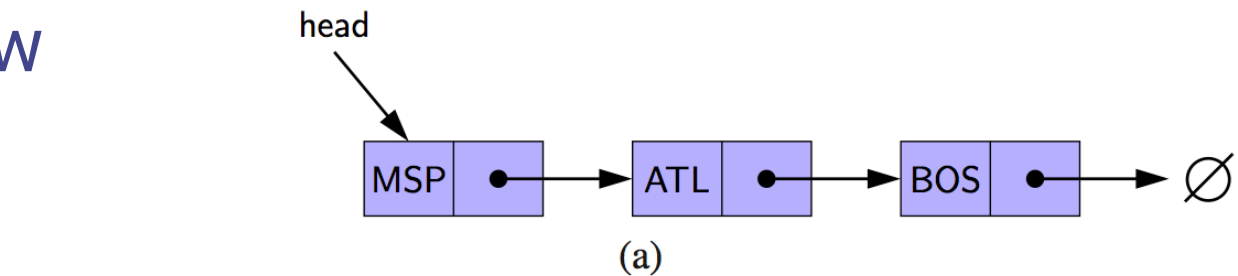
◆ Each node stores

- element
- link to the next node



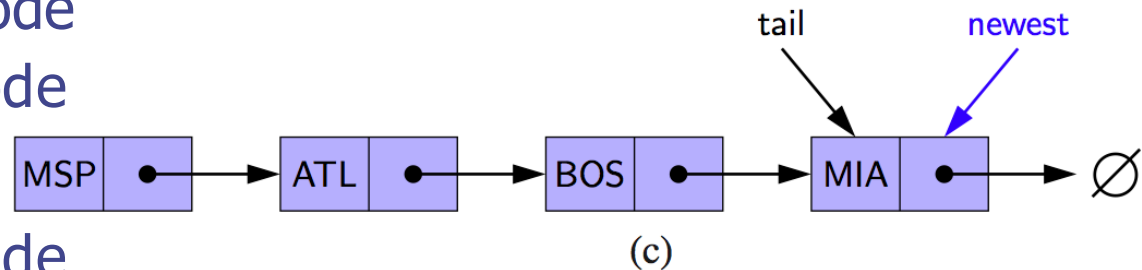
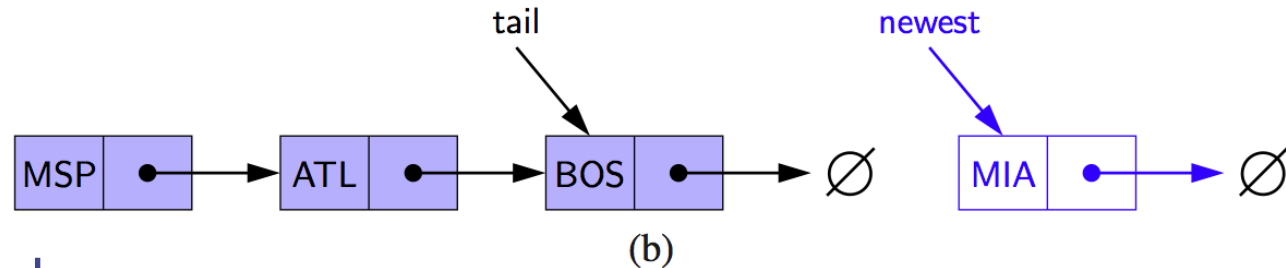
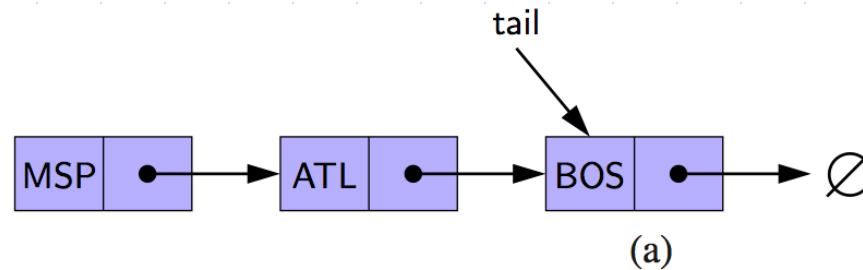
# Inserting at the Head

- Allocate new node
- Insert new element
- Have new node point to old head
- Update head to point to new node



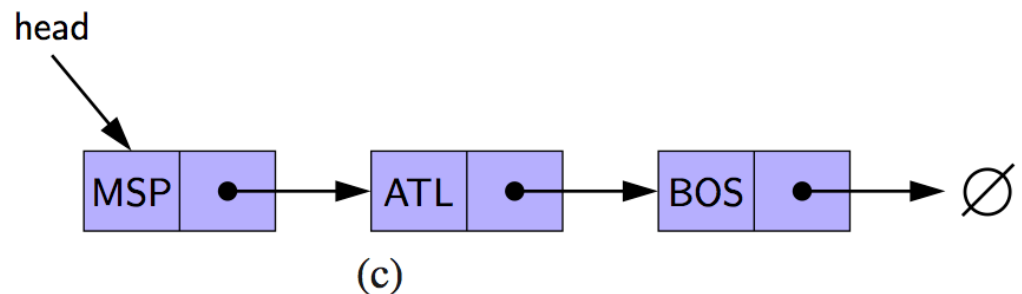
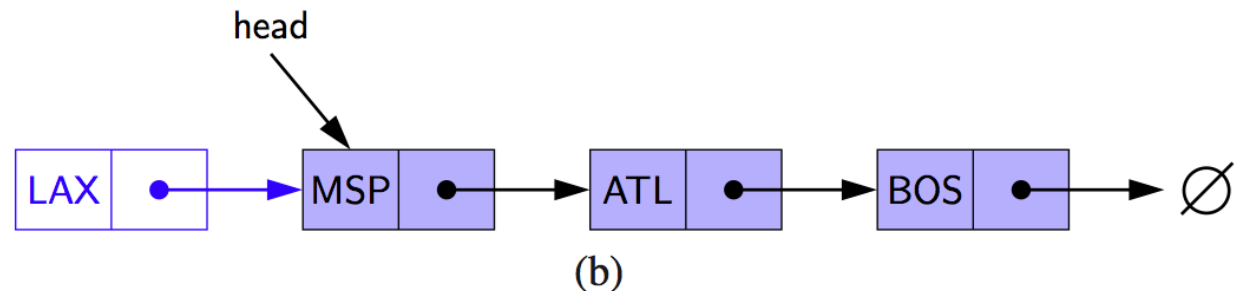
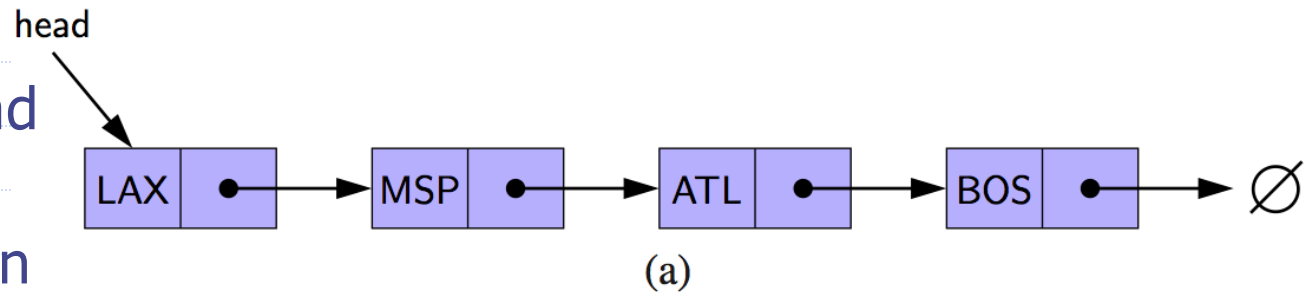
# Inserting at the Tail

- Allocate a new node
- Insert new element
- Have new node point to null
- Have old last node point to new node
- Update tail to point to new node



# Removing at the Head

- Update head to point to next node in the list
- Allow garbage collector to reclaim the former first node



# Removing at the Tail

- Removing at the tail of a singly linked list is not efficient!
- There is no constant-time way to update the tail to point to the previous node

