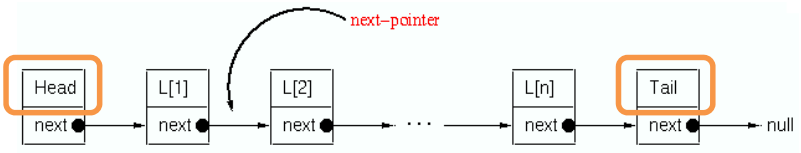
Lecture 3-2. Linked Lists

Sunghyun Cho
Professor
Division of Computer Science
chopro@hanyang.ac.kr

Keywords

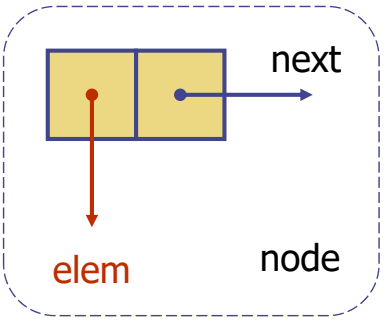
- Linked List
  - A representative type of list
  - Node, Pointer (Link), Head, Tail

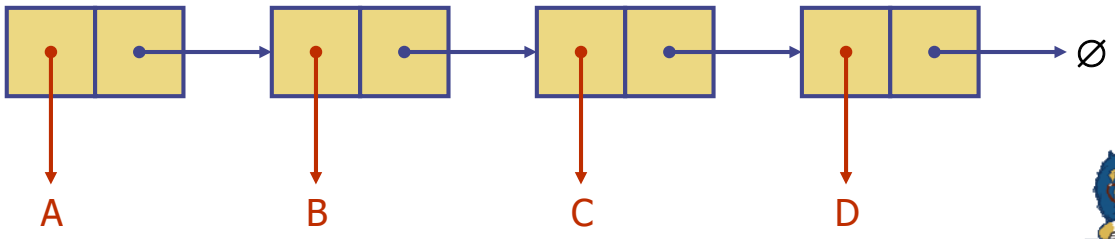


- Singly Linked List
  - Linked List with one directional pointer
- Doubly Linked List
  - Linked List with both directional pointers

Singly Linked List (§3.2)

- A singly linked list is a concrete data structure consisting of a sequence of nodes
- Each node stores
  - element
  - link to the next node





HANYANG UNIVERSITY

The Node Class for List Nodes

```
public class Node          {
    // Instance variables:
    private  Object element;
    private  Node next;
    /** Creates a node with null references to its element and next node. */
    public  Node()          {
       this(null,  null);
    }
    /** Creates a node with the given element and next node. */
    public  Node(Object e,  Node n)  {
         element  =  e;
         next  =  n;
    }
    // Accessor methods:
    public  Object getElement()  {
       return  element;
    }
    public  Node getNext()  {
       return  next;
    }
    // Modifier methods:
    public void  setElement(Object newElem)  {
         element  =  newElem;
    }
    public void  setNext(Node newNext)  {
         next  =  newNext;
    }
}
```

HANYANG UNIVERSITY

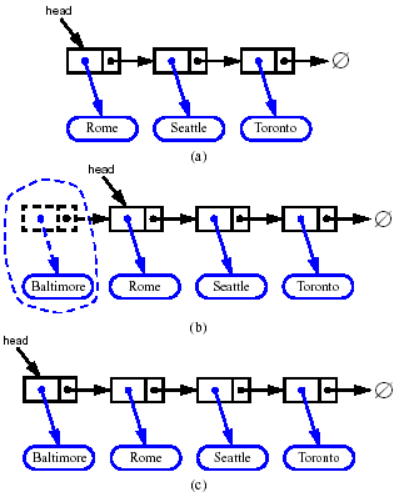
Inserting at the Head

1.

2.

3.

4.



HANYANG UNIVERSITY

Inserting at the Tail

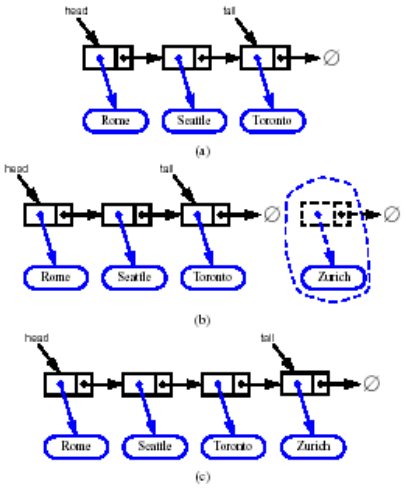
1.

2.

3.

4.

5.

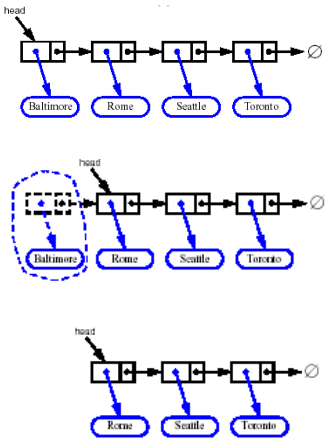


HANYANG UNIVERSITY

Removing at the Head

1.

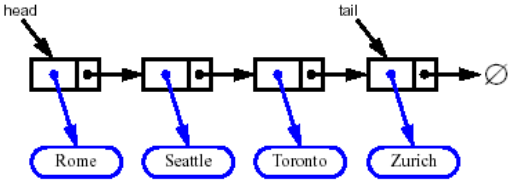
2.



Removing at the Tail

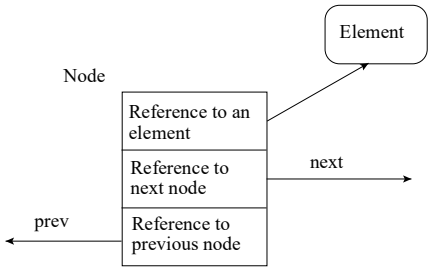
- **Removing at the tail of a singly linked list is not efficient!**
- There is no constant-time way to update the tail to point to the previous node

메모리 사용량도 빅O로 나타내면 링크드 리스트이던 어레이 리스트이든 O(n)가 된다



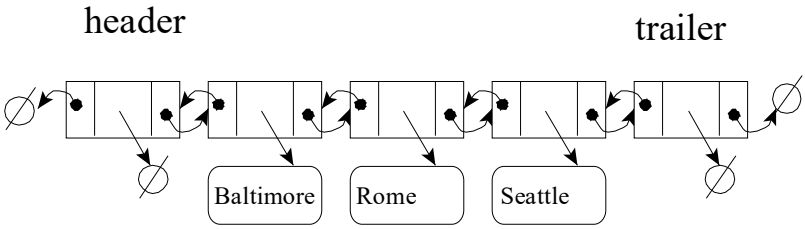
Doubly Linked List (§3.3)

- Recall that the deletion of an element at the tail is not easy because we have to find the node before the tail (the last node) by link hopping.
  - This problem can be easily solved by using the doubly linked list
- A **node** in a doubly linked list
  - A compound object that stores a reference to an element and two references, called **next** and **prev**, to the next and previous nodes, respectively.



Doubly Linked List

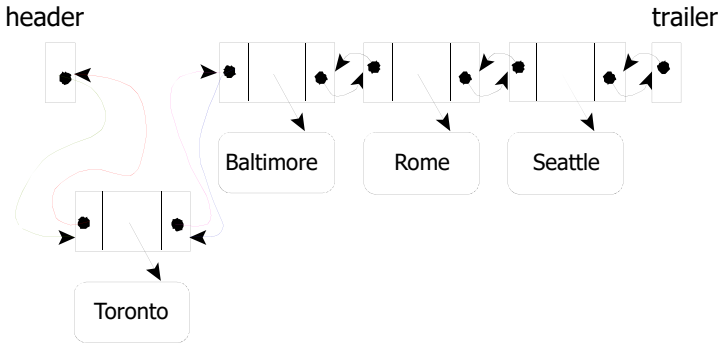
- Header and Trailer Sentinels
  - to simplify programming, a doubly linked list has a header and trailer nodes
  - they are called sentinel nodes which do not store any elements and just indicate both the ends of a list



- Difference from Singly Linked List
  - each node contains two links
  - two extra nodes: header and trailer, which contain no element

Insertion in the Middle of a Doubly Linked List

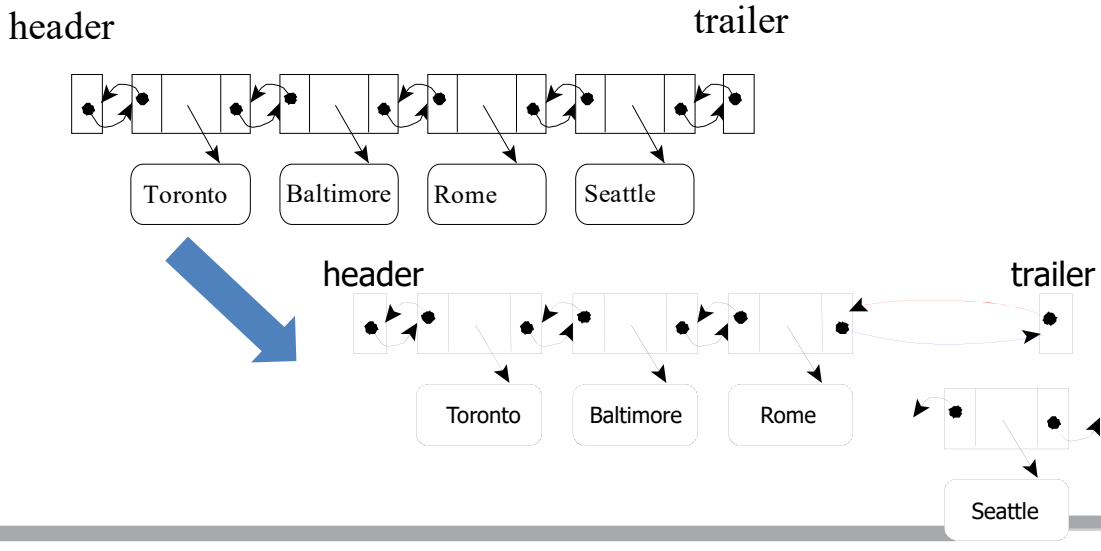
- Allowing for insertion and removal in the middle of the list
  - insert a new node ***z*** after ***v*** (***w*** is the next node of ***v***)
  1.
  2.
  3.
  4.



HANYANG UNIVERSITY

Removal in the Middle of a Doubly Linked List

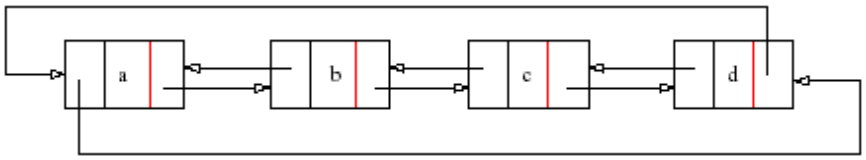
- Easy to remove a node ***v*** in the middle (even in the end) of a doubly linked list
  - to remove node ***v*** in the list, we simply have nodes, which are ***prev*** and ***next*** of node ***v***, point to each other



HANYANG UNIVERSITY

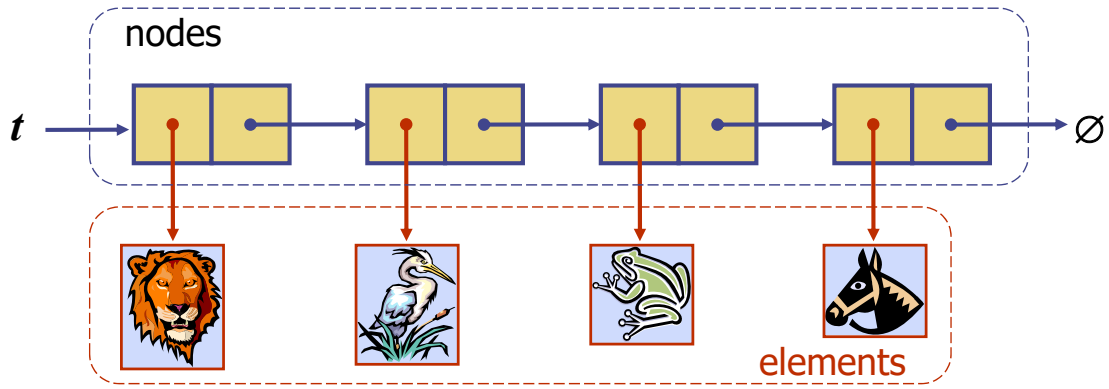
Circularly Linked List (§3.4)

- No Head (First) or Tail (Last) node
- Instead of having the last node's next pointer be **null**, it points back to the first node
  - if we traverse the nodes of a circularly linked list from any node by following next pointers, we will cycle through the nodes
- Cursor
  - marked as a special node which identifies the current position



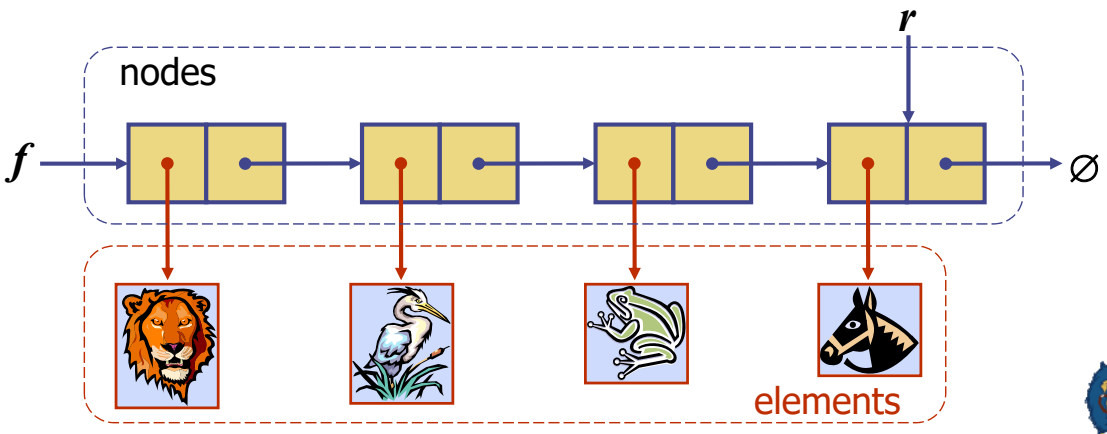
Stack as a Linked List (§ 5.1.3)

- We can implement a stack with a singly linked list
- The top element is stored at the first node of the list
- The space used is ($o(n)$) and each operation of the Stack ADT takes ($o(1)$) time



Queue as a Linked List (§ 5.2.3)

- We can implement a queue with a singly linked list
  - The front element is stored at the first node
  - The rear element is stored at the last node
- The space used is ( ) and each operation of the Queue ADT takes ( ) time



HANYANG UNIVERSITY