

# Linked Lists with Tail Pointers

Lecture 18

Section 18.1 - 18.3

Robb T. Koether

Hampden-Sydney College

Mon, Feb 26, 2018

1 Variations of Singly Linked Lists

2 Linked Lists with Tail Pointers

3 Assignment

# Outline

1 Variations of Singly Linked Lists

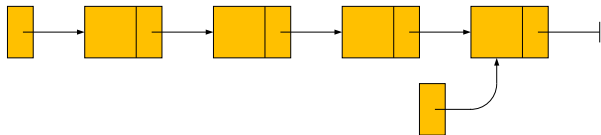
2 Linked Lists with Tail Pointers

3 Assignment

# Variations of Singly Linked Lists

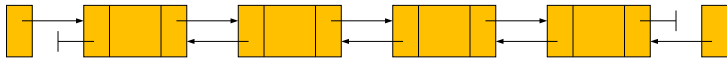
- There are many variations of the basic concept of a linked list.
  - Linked list with a tail pointer.
  - Doubly linked list.
  - Circularly linked list.

# Linked List with Tail Pointer



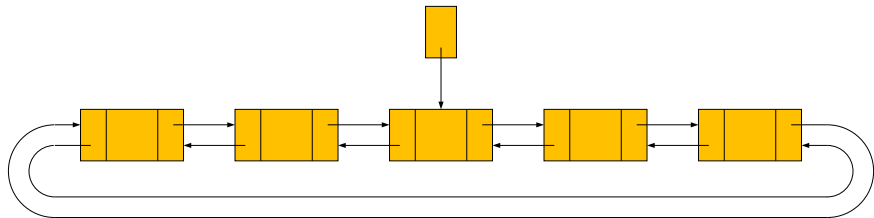
Linked list with a tail pointer

# Linked List with Tail Pointer



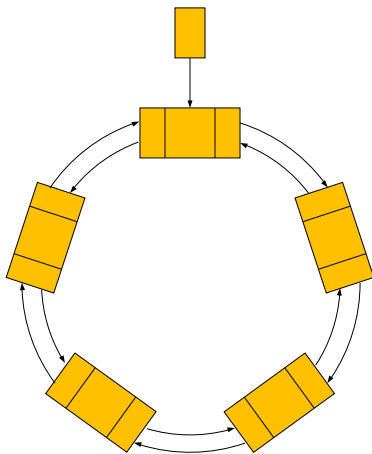
Doubly linked list

# Linked List with Tail Pointer



Circularly linked list

# Linked List with Tail Pointer



Circularly linked list



# Outline

1 Variations of Singly Linked Lists

**2 Linked Lists with Tail Pointers**

3 Assignment

# Linked Lists with Tail Pointers

## Definition (Linked List with Tail Pointer)

A **linked list with tail pointer** is a linked list with one additional pointer that points to the last node in the list.

## One Additional Data Member

`LinkedListNode* m_tail` - A pointer to the last node in the list.

- The name of the class is `LinkedListwTail`.

# Implementing the Member Functions

- The `LinkedListwTail` class is very similar to the `LinkedList` class.
- We need to rewrite only those functions that involve the tail pointer.
- `pushBack()` becomes much more efficient.
- Use a linked list with tail pointer in applications that make extensive use of `pushBack()`.

# Validity Requirements

- All the requirements of a `LinkedList`, plus
  - If `m_size == 0`, then `m_tail == NULL`.
  - If `m_size > 0`, then `m_tail` points to the last node.

# Outline

1 Variations of Singly Linked Lists

2 Linked Lists with Tail Pointers

**3 Assignment**

# Assignment

## Assignment

- Read Sections 18.1 - 18.3.
- Also, you can google “linked list with tail pointer” and find a number web sites that discuss this.