

DOUBLY LINKED LISTS



Doubly Linked List

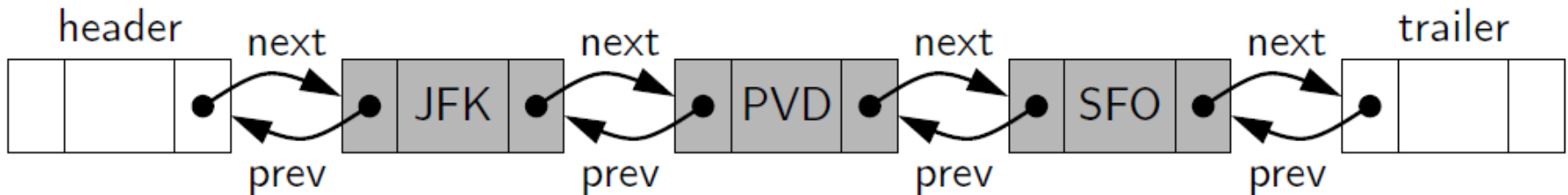
- Doubly Linked List Overview
- Header and Trailer Sentinels

Doubly Linked List

- In a doubly linked list, we define a linked list in which each node keeps an explicit reference to the node before it **and** a reference to the node after it.
- These lists allow a greater variety of $O(1)$ -time update operations, including insertions and deletions.
- We continue to use the term “next” for the reference to the node that follows another.
- We have a new term “prev” for the reference to the node that precedes it.

Sentinel Node

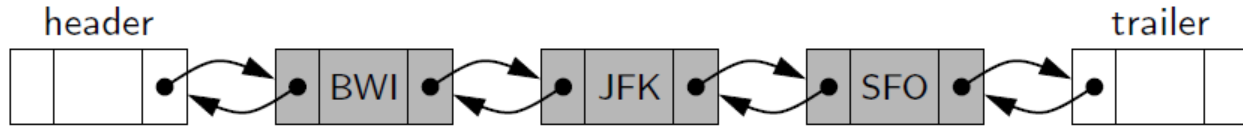
- We add special nodes at both ends of the list.
- a **header** node at the beginning of the list
- a **trailer** node at the end of the list.
- These “dummy” nodes are known as



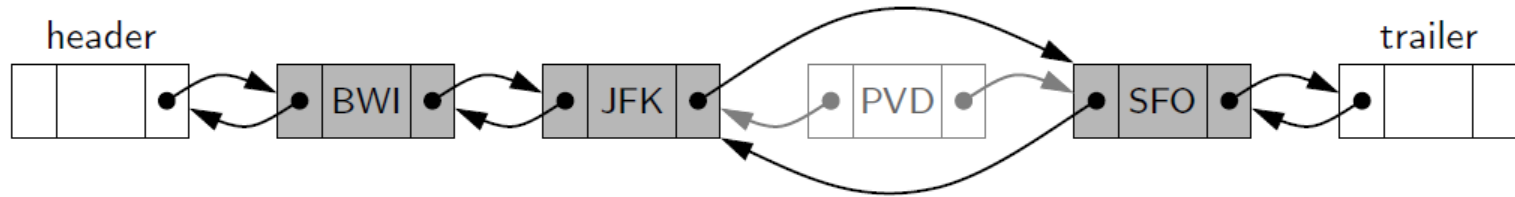
Inserting and Deleting with a Doubly Linked List

- Every insertion into our doubly linked list representation will take place between a pair of existing nodes
- When a new element is inserted at the front of the sequence, we will simply add the new node *between* the header and the node that is currently after the header.

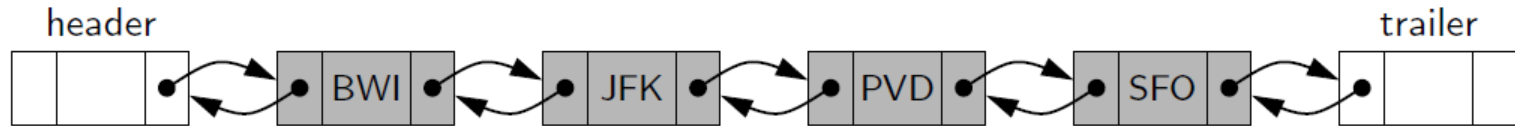
Inserting and Deleting with a Doubly Linked List



(a)



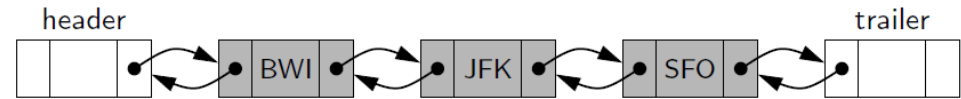
(b)



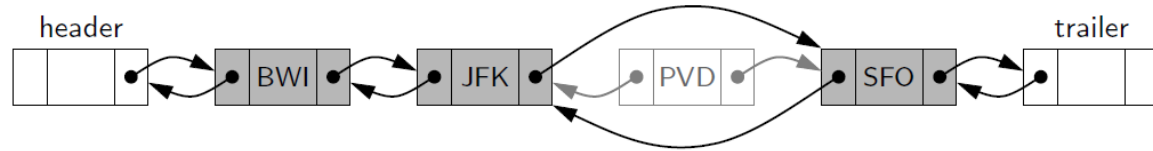
(c)

Inserting and Deleting with a Doubly Linked List

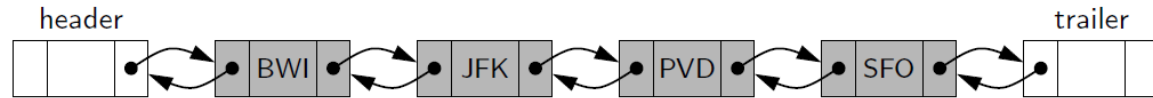
- (a) before the operation
- (b) after creating the new node
- (c) after linking the neighbors to the new node



(a)

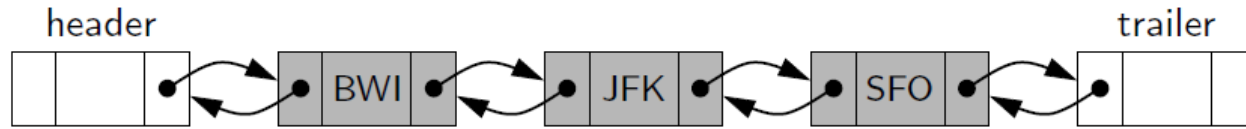


(b)

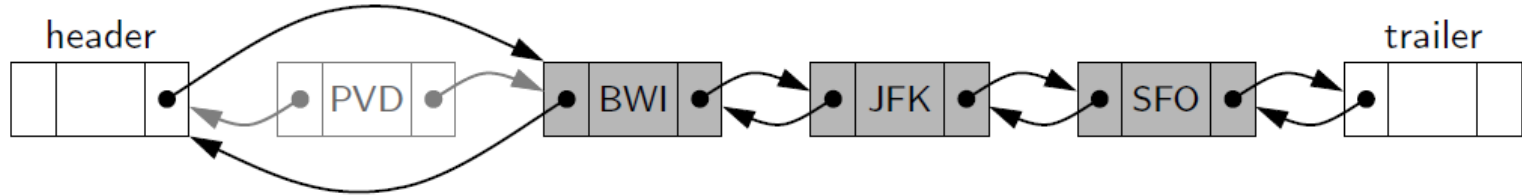


(c)

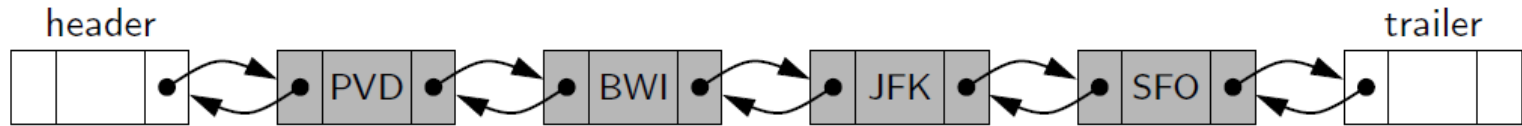
Insertion of a Node to Front



(a)



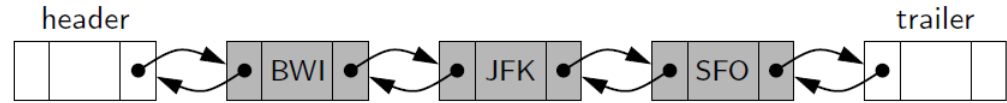
(b)



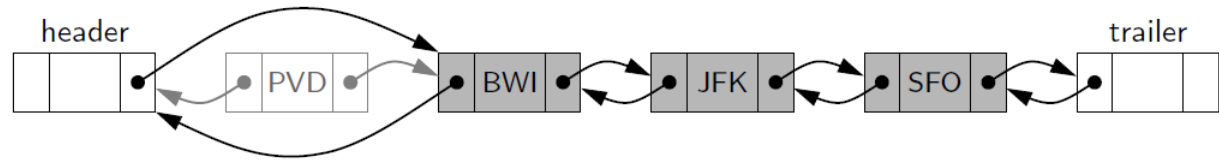
(c)

Insertion of a Node to Front

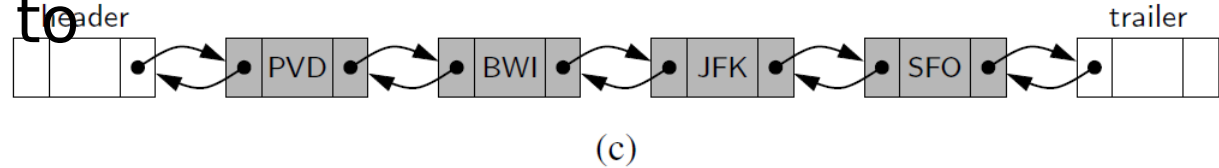
- (a) before the operation



- (b) after creating the new node



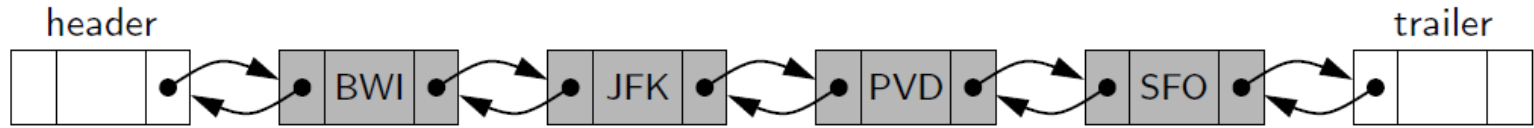
- (c) after linking the neighbors to the new node



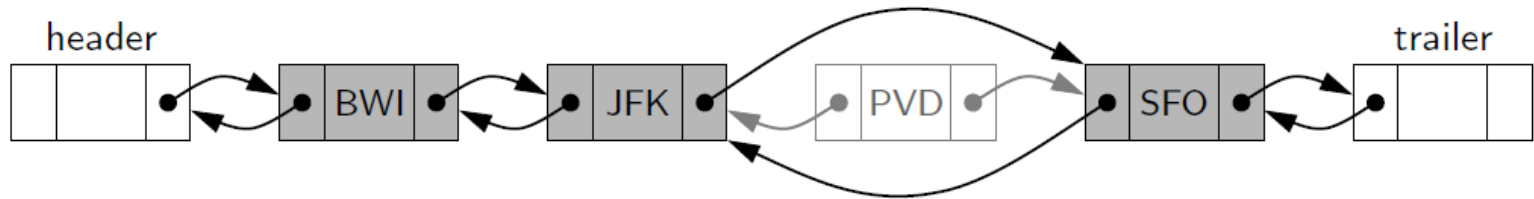
Deletion of a Node

- The two neighbors of the node to be deleted are linked directly to each other
- As a result, that node will no longer be considered part of the list and it can be reclaimed by the system.
- Because of sentinels, the same implementation can be used when deleting the first or the last element of a sequence.

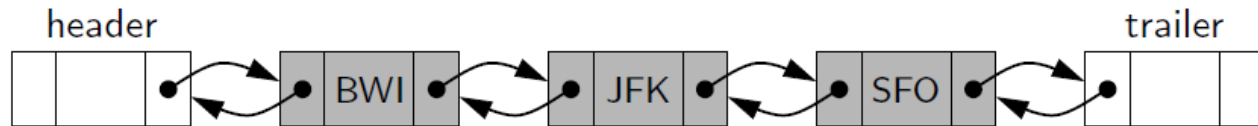
Deletion of a Node



(a)



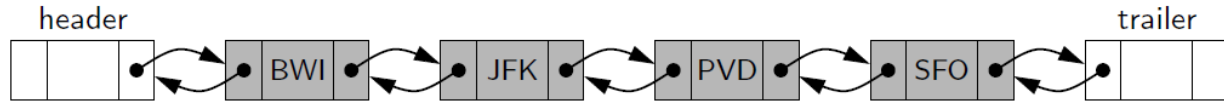
(b)



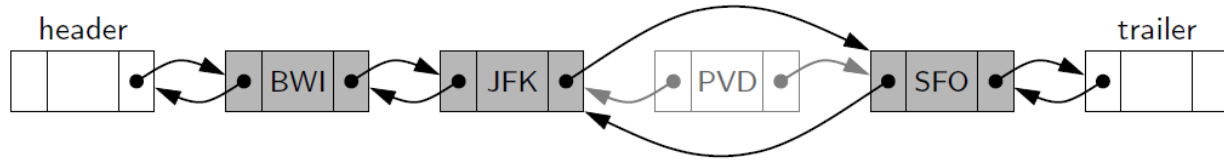
(c)

Deletion of a Node

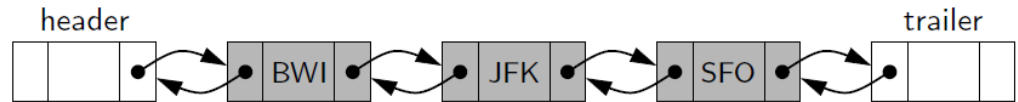
- (a) before the removal
- (b) after linking out the old node
- (c) after the removal



(a)



(b)



(c)

Review

- Learned about Doubly Linked List Nodes
- Insertion
- Deletion
- Up next, implementation of a Doubly Linked List!