

Tarea 14 Listas Doblemente enlazadas

Renteria Magaña Rayni Damian 17212178

To provide greater symmetry, 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. Such a structure is known as a **doubly linked list**. These lists allow a greater variety of $O(1)$ -time update operations, including insertions and deletions at arbitrary positions within the list. We continue to use the term “**next**” for the reference to the node that follows another, and we introduce the term “**prev**” for the reference to the node that precedes it.

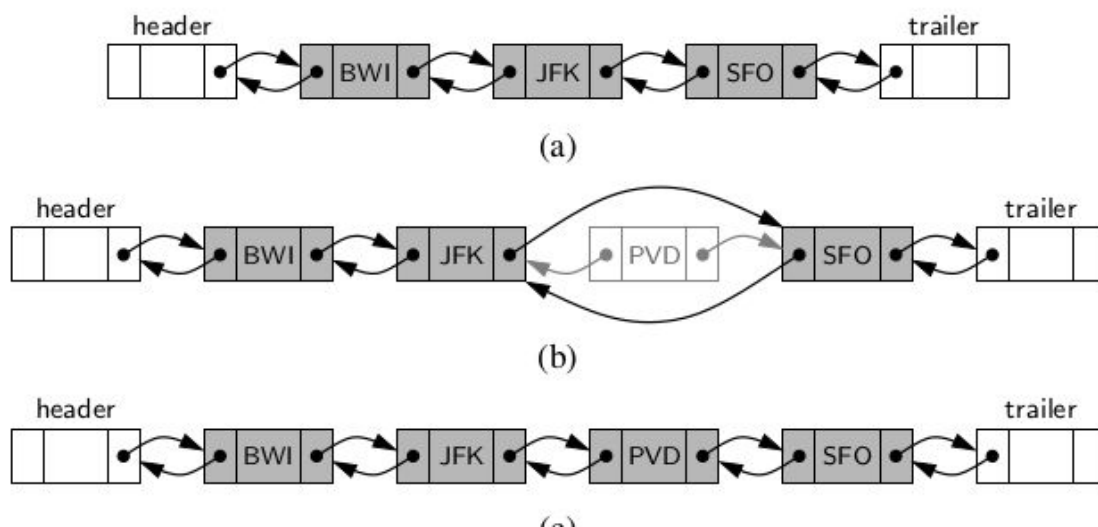
Header and Trailer Sentinels

In order to avoid some special cases when operating near the boundaries of a doubly linked list, it helps to add special nodes at both ends of the list: a header node at the beginning of the list, and a trailer node at the end of the list. These “dummy” nodes are known as sentinels (or guards), and they do not store elements of the primary sequence.

When using sentinel nodes, an empty list is initialized so that the next field of the header points to the trailer, and the prev field of the trailer points to the header; the remaining fields of the sentinels are irrelevant (presumably None, in Python). For a nonempty list, the header’s next will refer to a node containing the first real element of a sequence, just as the trailer’s prev references the node containing the last element of a sequence.

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.



References

Goodrich, M. T., Goldwasser, M. H., & Tamassia, R. (2013). *Data structures and algorithms in Python*. Wiley.