

unsorted linked lists. An unsorted linked list can be sorted. We shall see in this chapter that some (but not all) searching, merging, and sorting algorithms can be adapted to linked lists.

## 4.2 Insertion

In this section we shall study algorithms for inserting a new node at some point in a linked list. There are potentially four different cases to be handled:

- (1) Insertion in an empty linked list.
- (2) Insertion before the first node of a nonempty linked list.
- (3) Insertion after the last node of a nonempty linked list.
- (4) Insertion between nodes of a nonempty linked list.

In general, the insertion algorithm must be given (or must obtain for itself) links to one or both of the new node's future neighbors, in order to allow the new node to be linked properly.

### 4.2.1 Insertion in a singly-linked list

Algorithm 4.11 inserts a new node at some point in the SLL headed by *first*. Step 1 creates the new node. What follows depends on where the new node is to be inserted.

Step 2 inserts the new node before the first node. The new node's successor becomes *first*, and *first* itself becomes a link to the new node. Step 2 handles case (2) above; fortuitously it also handles case (1), if *first* happens to be null.

Step 3 inserts the new node after some indicated node, given a link *pred* to the indicated node. The new node's successor is set to the former successor of node *pred*, and node *pred*'s successor becomes the new node. Step 3 handles case (4) above; fortuitously it also handles case (3), if *pred* happens to be a link to the last node.

Figure 4.12 illustrates the SLL insertion algorithm's behavior.

To insert *elem* at a given point in the SLL headed by *first*:

1. Make *ins* a link to a newly-created node with element *elem* and successor null.
2. If the insertion point is before the first node:
  - 2.1. Set node *ins*'s successor to *first*.
  - 2.2. Set *first* to *ins*.
3. If the insertion point is after the node *pred*:
  - 3.1. Set node *ins*'s successor to node *pred*'s successor.
  - 3.2. Set node *pred*'s successor to *ins*.
4. Terminate.

**Algorithm 4.11** SLL insertion algorithm.