CS3000: Algorithms & Data — Summer '25 — Laney Strange

Linked List Pseudocode

Below is the pseudocode for the key operations of a Linked List data structure we covered in class. There are many variations on Linked Lists; this one is doubly-linked and includes a head pointer but no tail pointer. The procedures below implement insert, find, and remove. Insert runs in constant time, find runs in linear time, and remove (given the node we wish to remove) runs in constant time.

LIST-INSERT(L, x)

```
1  x.next = L.head
2  x.prev = NIL
3  if L.head ≠ NIL
4        L.head.prev = x
5  L.head = x
```

LIST-SEARCH(L, k)

```
1  x = L.head
2  while x ≠ NIL and x.key ≠ k
3        x = x.next
4  return x
```

LIST-DELETE(L, x)

```
1  if x.prev ≠ NIL
2        x.prev.next = x.next
3  else
4        L.head = x.next
5  if x.next ≠ NIL
6        x.next.prev = x.prev
```

We typeset the procedures above with the following LaTeX:

```
\begin{codebox}
\Procname{$\proc{List-Insert}(L, x)$}
\li $x.\id{next} \gets L.\id{head}$
\li $x.\id{prev} \gets \const{Nil}$
\li \If $L.\id{head} \ne \const{Nil}$
\Then
\li $L.\id{head}.\id{prev} \gets x$
\End
\li $L.\id{head} \gets x$
\end{codebox}

\begin{codebox}
```

```
\Procname{$\proc{List-Search}(L, k)$}
\li $x \gets L.\id{head}$
\li \While $x \ne \const{Nil}$ and $x.\id{key} \ne k$
\Do
\li $x \gets x.\id{next}$
\End
\li \Return  $x$
\end{codebox}

\begin{codebox}
\Procname{$\proc{List-Delete}(L, x)$}
\li \If $x.\id{prev} \ne \const{Nil}$
\Then
\li $x.\id{prev}.\id{next} \gets x.\id{next}$
\End
\li \Else
\Do
\li $L.\id{head} \gets x.\id{next}$
\End
\li \If $x.\id{next} \ne \const{Nil}$
\Then
\li $x.\id{next}.\id{prev} \gets x.\id{prev}$
\End
\end{codebox}
```