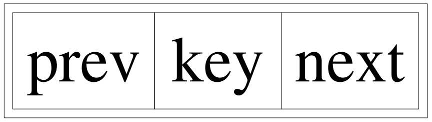
Doubly Linked Lists in Practice

As in the Book

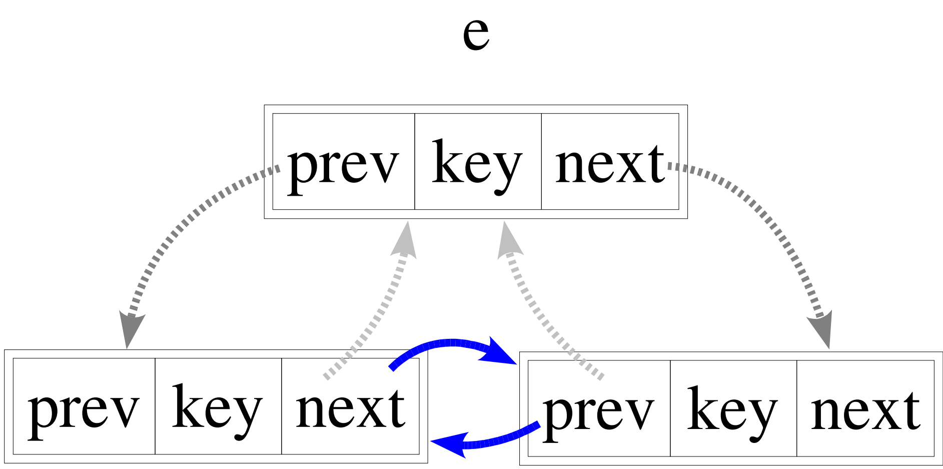
- Pointers to previous and next:

```
struct elem_t {
    struct elem_t *prev, *next;
    datatype_t key;
};
```

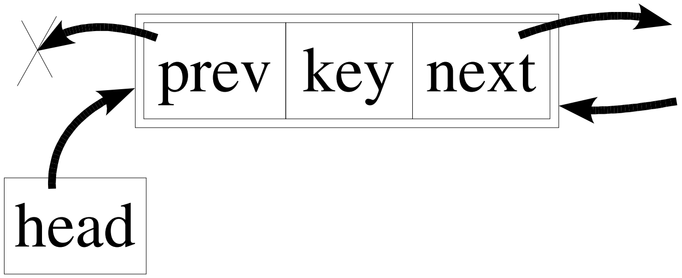


- Remove(e) (simplified):

```
e->prev->next=e->next
e->next->prev=e->prev
```



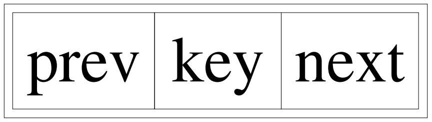
- Problem for the head because it is not a struct elem_t.



Often in Practice

- Different pointers:

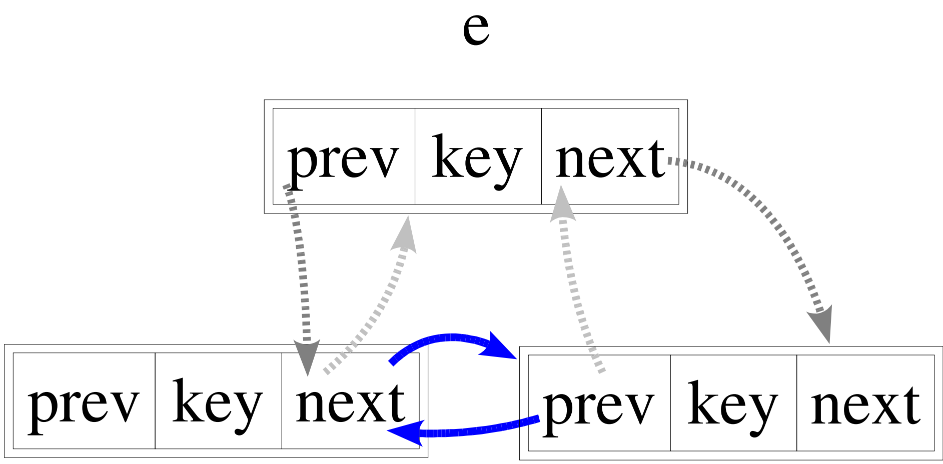
```
struct elem_t {
    struct elem_t **prev, *next;
    datatype_t key;
};
```



prev key next

- Remove(e) (simplified):

```
*e->prev=e->next
e->next->prev=e->prev
```



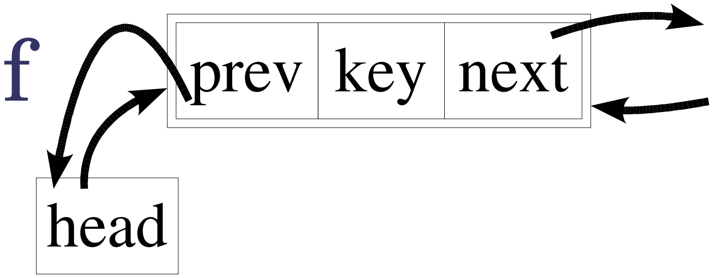
e

prev key next

prev key next

prev key next

- No special case for the head of the list.



prev key next

head