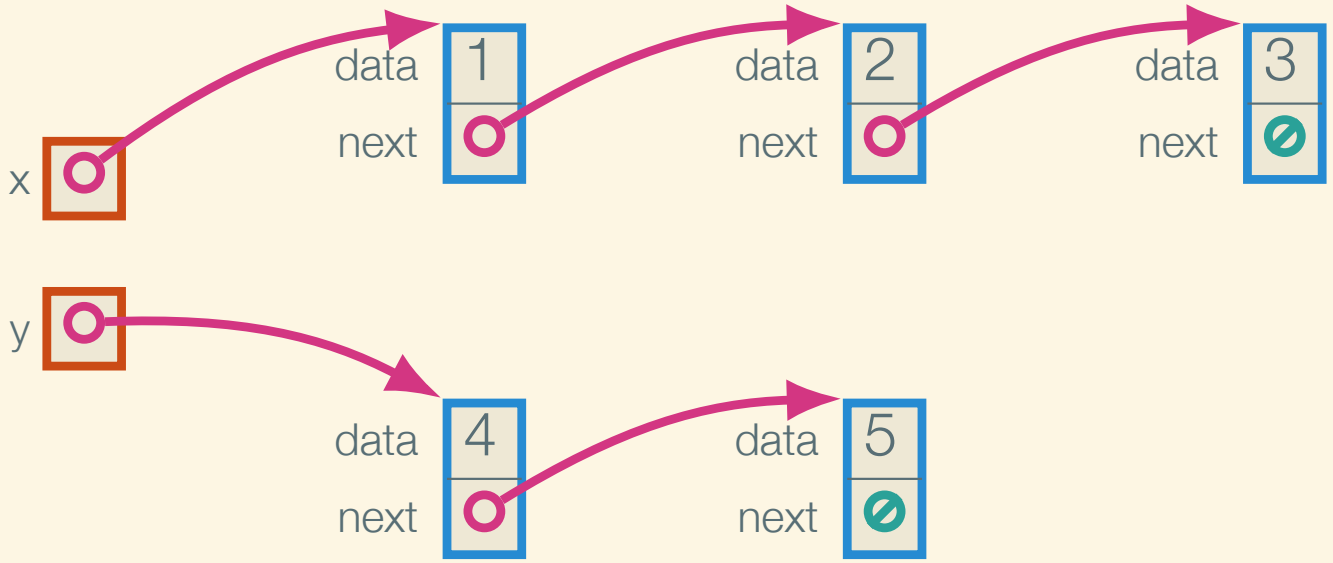
Linked Data Structures

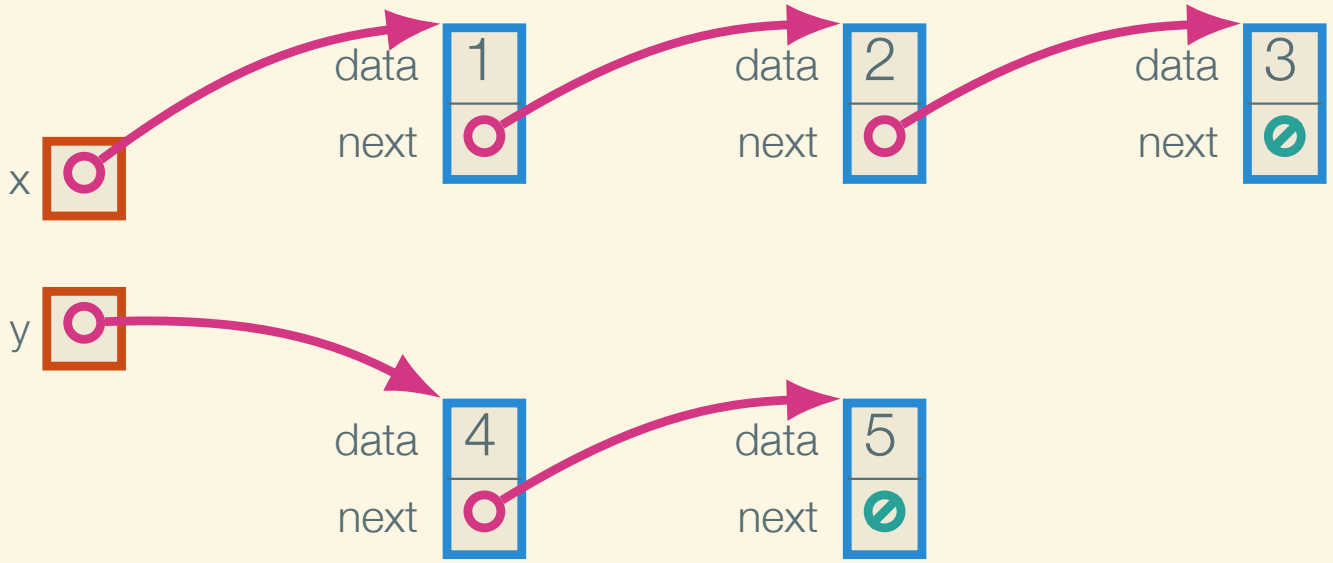
EECS 211

Winter 2017

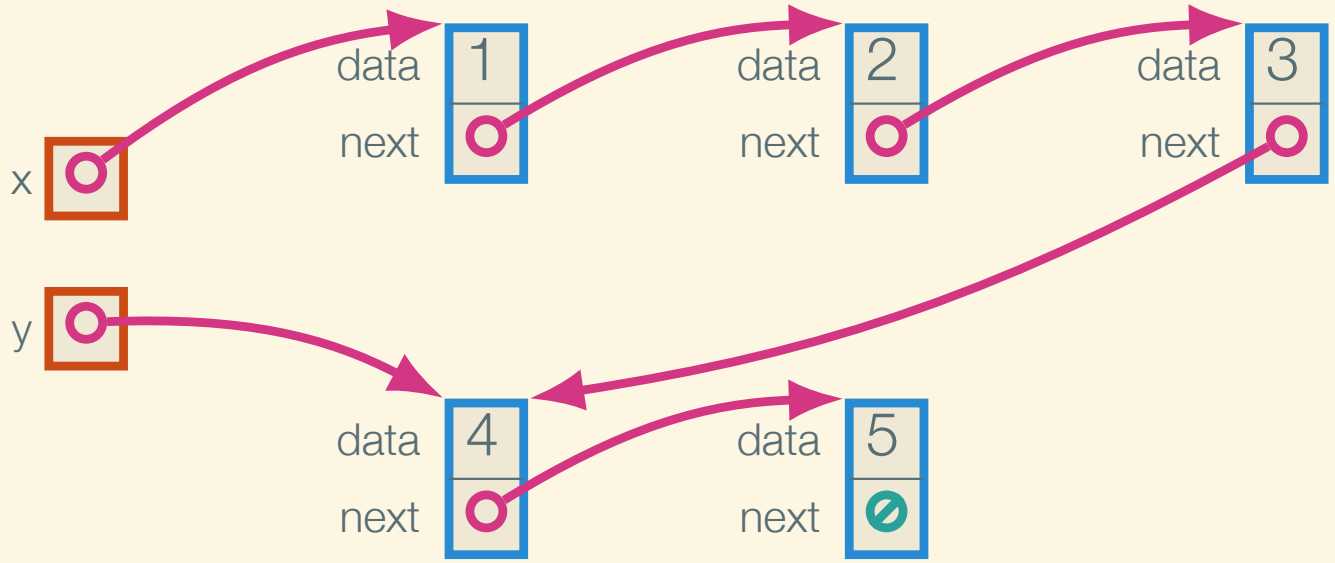
```
List x = cons("1", cons("2", cons("3", nullptr)));
List y = cons("4", cons("5", nullptr));
```



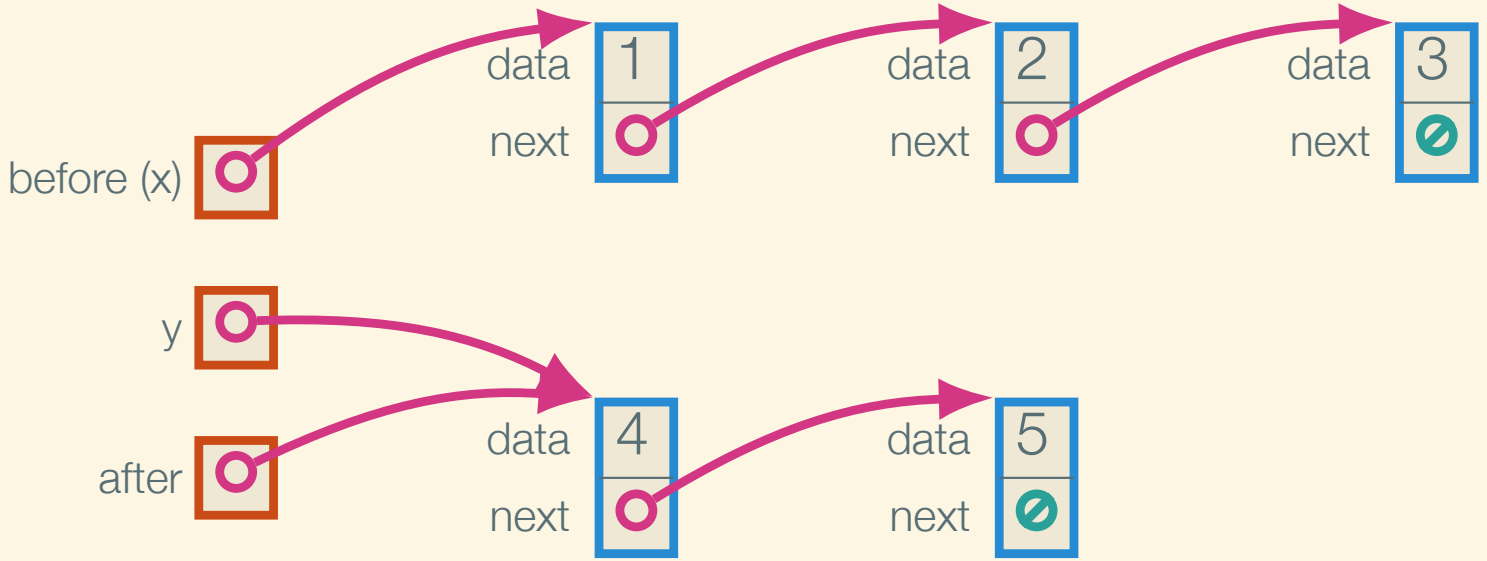
```
List x = cons("1", cons("2", cons("3", nullptr)));
List y = cons("4", cons("5", nullptr));

concat(x, y);
```



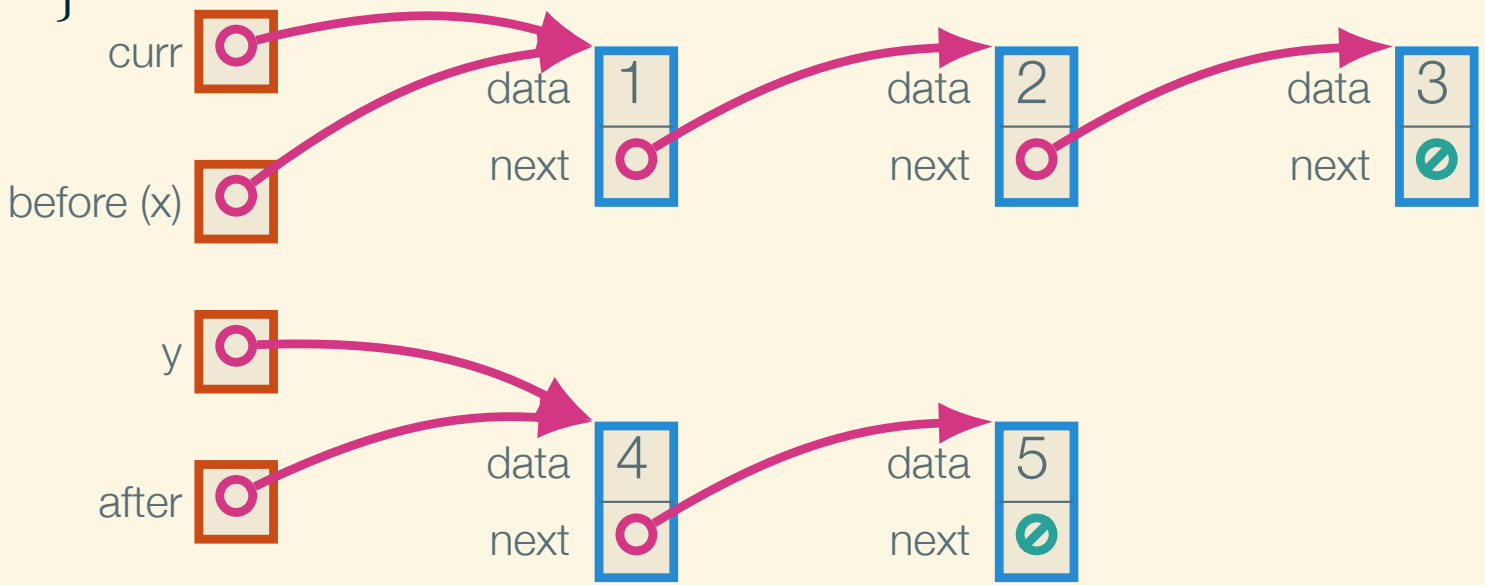
```
List x = cons("1", cons("2", cons("3", nullptr)));
List y = cons("4", cons("5", nullptr));

concat(x, y);
```



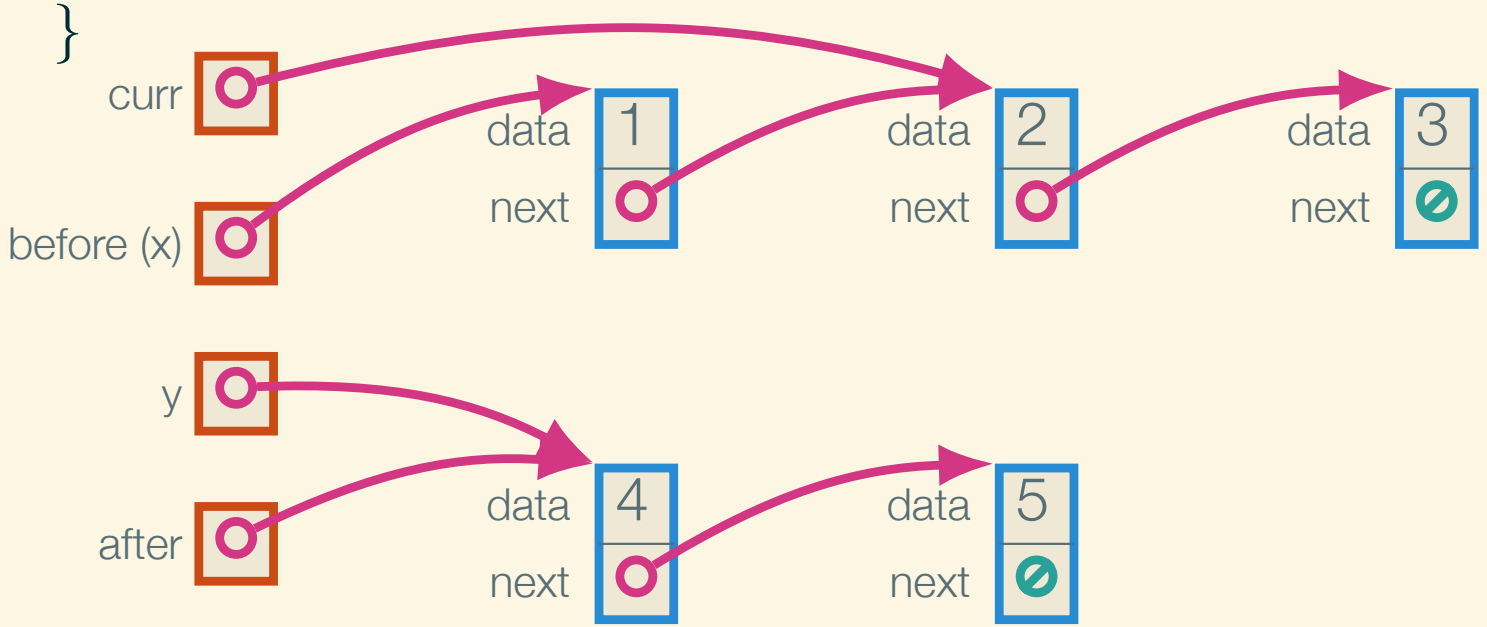
```
void concat(List& before, List after) {
    if (before == nullptr) before = after;
    else {
        List curr = before;
        while (curr→next != nullptr) curr = curr→next;
        curr→next = after;
    }
}
```



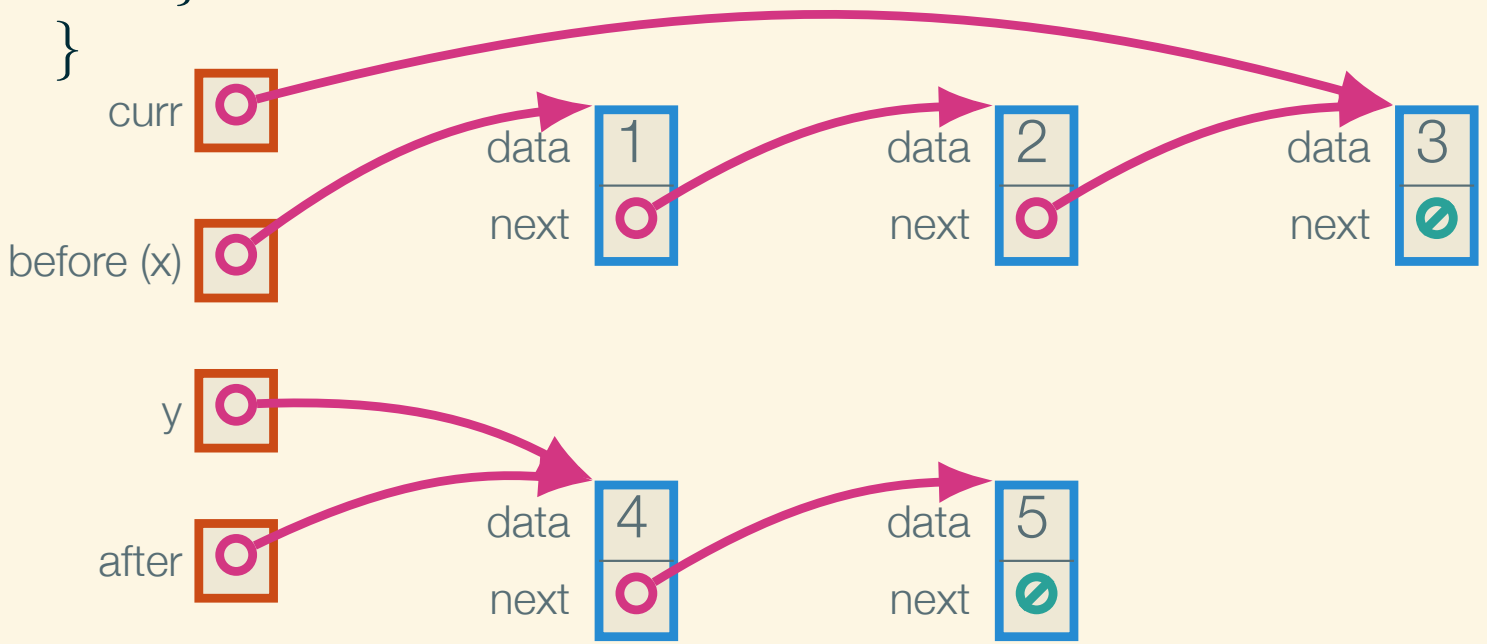
```
void concat(List& before, List after) {
    if (before == nullptr) before = after;
    else {
        List curr = before;
        while (curr→next != nullptr) curr = curr→next;
        curr→next = after;
    }
}
```



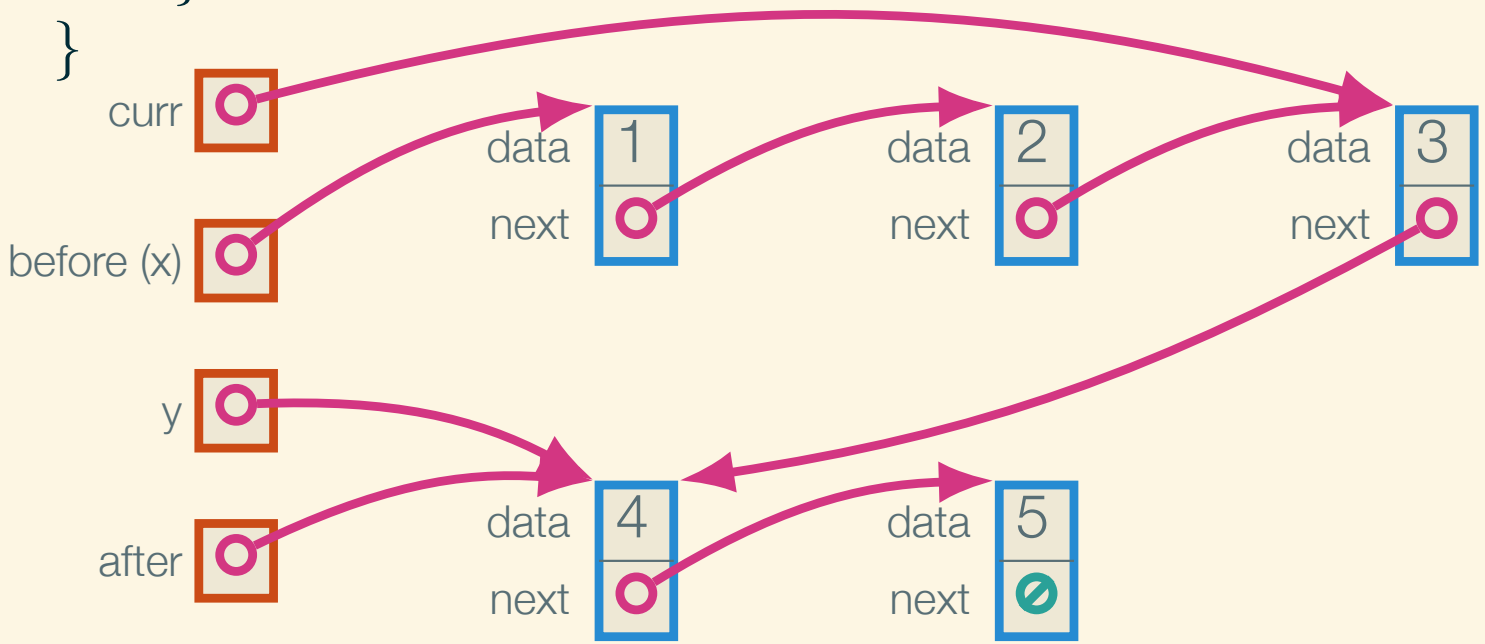
```
void concat(List& before, List after) {
    if (before == nullptr) before = after;
    else {
        List curr = before;
        while (curr→next != nullptr) curr = curr→next;
        curr→next = after;
    }
}
```



```
void concat(List& before, List after) {
    if (before == nullptr) before = after;
    else {
        List curr = before;
        while (curr→next != nullptr) curr = curr→next;
        curr→next = after;
    }
}
```



```
void concat(List& before, List after) {
    if (before == nullptr) before = after;
    else {
        List curr = before;
        while (curr→next != nullptr) curr = curr→next;
        curr→next = after;
    }
}
```



```
void concat(List& before, List after) {
    if (before == nullptr) before = after;
    else {
        List curr = before;
        while (curr→next != nullptr) curr = curr→next;
        curr→next = after;
    }
}
```