

Antworten zu Übungsblatt Nr. 7

Aufgabe 2

Anzahl der Blockoperationen im schlechtesten Fall für reverse,
n (Anzahl gespeicherter Elemente)
B (Blockgröße)
M (Größe des schnellen speichers)

reverse

Worst case: jedes Element der LinkedList ist in einem anderen Block verteilt über den schnellen speicher, wodurch wir bereits n Blockoperationen haben. Wenn außerdem unsere LinkedList selber auf einem anderen Block ist brauchen wir für reverse mindestens $n + 1$ Blockoperationen.

splice

Worst case: Klassendefinition in anderem Block als Information, somit bereits 2

```
// -----
LinkedList LinkedList::splice(LinkedListItem* begin,
                             LinkedListItem* end) {
    if (begin != firstItem) {
        begin->previousItem->nextItem = end->nextItem; // plus 3 (worst)
        // ^           ^           ^ required
    } else {
        firstItem = end->nextItem;
    }
    if (end != lastItem) {
        end->nextItem->previousItem = begin->previousItem; // plus 1 (worst, anderen
                                                           // beiden bereits geladen)
        // ^           ^ required, anderen beiden sind bereits geladen
    } else {
        lastItem = begin->previousItem;
    }

    // bereits geladen
    begin->previousItem = NULL;
    end->nextItem = NULL;

    // plus 1
    LinkedList list;
```

```
    list.firstItem = begin;
    list.lastItem = end;

    return list;
    // total: 7 Blockoperationen im worst case.
    // meist weniger.
}
```

Damit wären wir bei $\theta(n)$ für reverse und $\theta(1)$ für splice.
(da konstante faktoren)