

## Übungsblatt 7

Abgabe bis Dienstag, den 20. Juni um 12:00 Uhr

### Aufgabe 1 (10 Punkte)

Erweitern Sie die Klasse *LinkedList* aus der Vorlesung um die folgenden beiden Operationen. Achten Sie darauf, dass die Operationen die jeweils angegebene Laufzeit haben. Auf dem Wiki finden Sie eine TIP Datei mit einem Unit Test für jede Operation. Schreiben Sie darüber hinaus noch jeweils mindestens einen Unit Test für einen Spezialfall. Die korrekte Behandlung von Spezialfällen ist bei verketteten Listen besonders kritisch.

*reverse()*: Umdrehen der Liste, so dass, wenn man bei *first* anfängt und dann den *next* Zeigern folgt, man die Elemente gerade in umgekehrter Reihenfolge bekommt wie vorher. Diese Operation sollte Laufzeit  $O(n)$  haben, wenn  $n$  Elemente in der Liste sind.

*splice( $l_1$ ,  $l_2$ )*: Ausschneiden der Teilliste, die bei  $l_1$  beginnt und bei  $l_2$  endet (beide vom Typ *LinkedListItem*). Die Operation soll die Teilliste zurückgeben und die ursprüngliche Liste sollte die Elemente aus der Teilliste nicht mehr enthalten. Diese Operation sollte Laufzeit  $O(1)$  haben. Sie dürfen dabei annehmen, dass  $l_1$  und  $l_2$  beide in der Liste vorkommen und  $l_2$  nicht vor  $l_1$  kommt.

*Zusatzdenkaufgabe*: Wie könnte man die Klasse erweitern, so dass *reverse()* in Zeit  $O(1)$  läuft?

### Aufgabe 2 (10 Punkte)

Bestimmen Sie die Anzahl der Blockoperationen im schlechtesten Fall für jede der beiden Operationen aus Aufgabe 1 (*reverse* und *splice*). Diese Anzahl kann dabei von folgenden Parametern abhängen:  $n$  (Anzahl gespeicherter Elemente),  $B$  (Blockgröße),  $M$  (Größe des schnellen Speichers). Bitte nicht raten sondern nachdenken.

Committed Sie Ihren Code für Aufgabe 1 (als Code) und Ihre Lösung für Aufgabe 2 (als PDF) in unser SVN, in einen neuen Unterordner *blatt-07*. Alles andere wie gehabt, inklusive einer aussagekräftigen *erfahrungen.txt* Datei.

Warum ist unser Gehirn so groß und warum nicht noch größer?