

Algorithmen und Datenstrukturen

Master

Listen

Inhalt

- Terminologie
- Einfach verkettete Listen
- Doppelt verkettete Listen

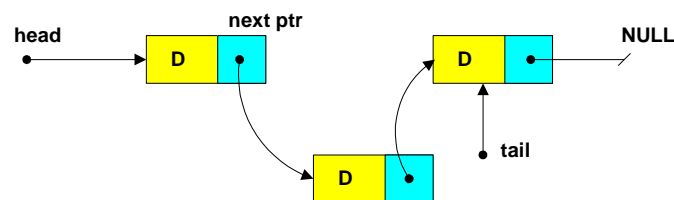
Terminologie

- ▶ Eine Liste besteht aus Knoten (nodes), welche zwei Elemente haben:
 - Daten
 - Verweis auf das nächste Element
- ▶ Zur Liste gehört ein Head-Knoten (der Beginn der Liste) und meist auch ein Tail-Knoten
- ▶ Letzter Verweis zeigt auf „null“

Verkettete Liste

- ▶ Beliebige viele Elemente
- ▶ Jedes Element hat 2 Teile
 - Nutzdaten
 - Zeiger auf das nächste Element (next)
- ▶ Ende der Liste wenn `next == NULL`
- ▶ Elemente liegen irgendwo im Speicher, Zeiger ist daher WICHTIG!

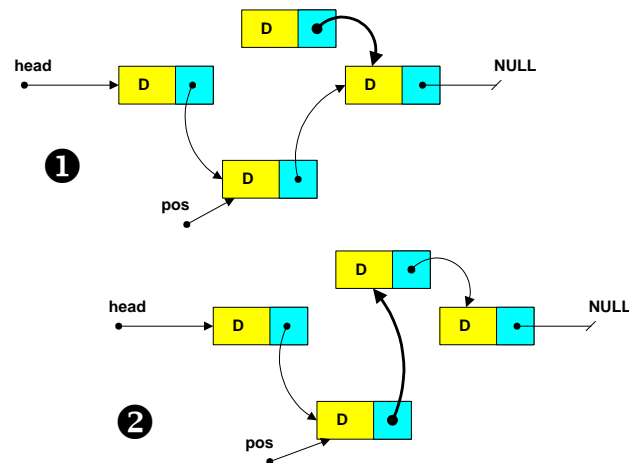
Verkettete Liste



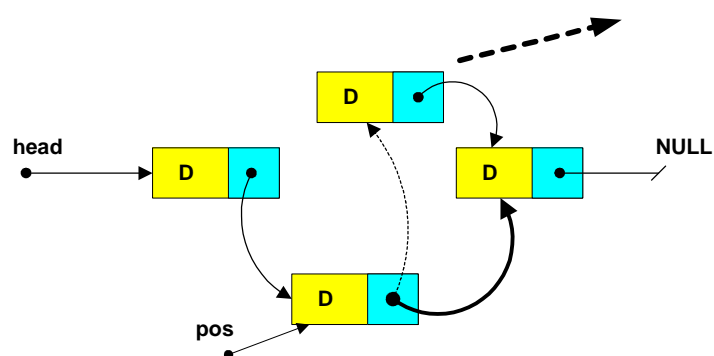
Verkettete Liste

- ▶ Einfaches Einfügen und Löschen $O(1)$
 - Pro Operation werden nur 2 Zeiger „verbogen“
 - große Nutzdatenmengen müssen nicht im Speicher bewegt werden
- ▶ Zugriff nur sequentiell in eine Richtung möglich, d.h. Suchen mit $O(n)$

Verkettete Liste – Einfügen



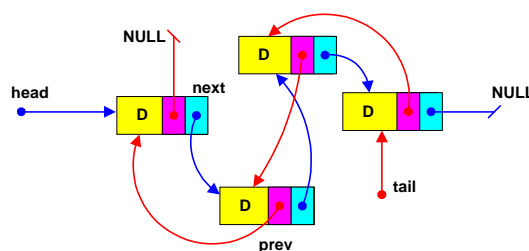
Verkettete Liste – Löschen



Doppelt verkettete Liste

- ▶ Wie einfach verkettete Liste
- ▶ Jedes Element hat 3 Teile
 - Nutzdaten
 - Zeiger auf nächstes Element (next)
 - Zeiger auf voriges Element (prev)
- ▶ Zugriff sequentiell, aber jetzt vor und zurück

Doppelt verkettete Liste



Iterator

- ▶ Ein Iterator gestattet es, unabhängig von der Liste eine oder mehrere Positionen zu referenzieren
- ▶ Iterator bietet Operationen Next() und Previous(), um in der Liste hin- und herzuwandern

Iterator

- ▶ Liste muss um Methoden ergänzt werden, die Iterator Objekt akzeptieren
 - InsertBefore
 - InsertAfter
 - RemoveAt

Aufgabe 5

- ▶ Modifiziere die Liste derart, dass eine doppelt verkettete Liste entsteht.
- ▶ Schreibe einen Iterator, der das Durchlaufen der Liste in die verkehrte Richtung gestattet.