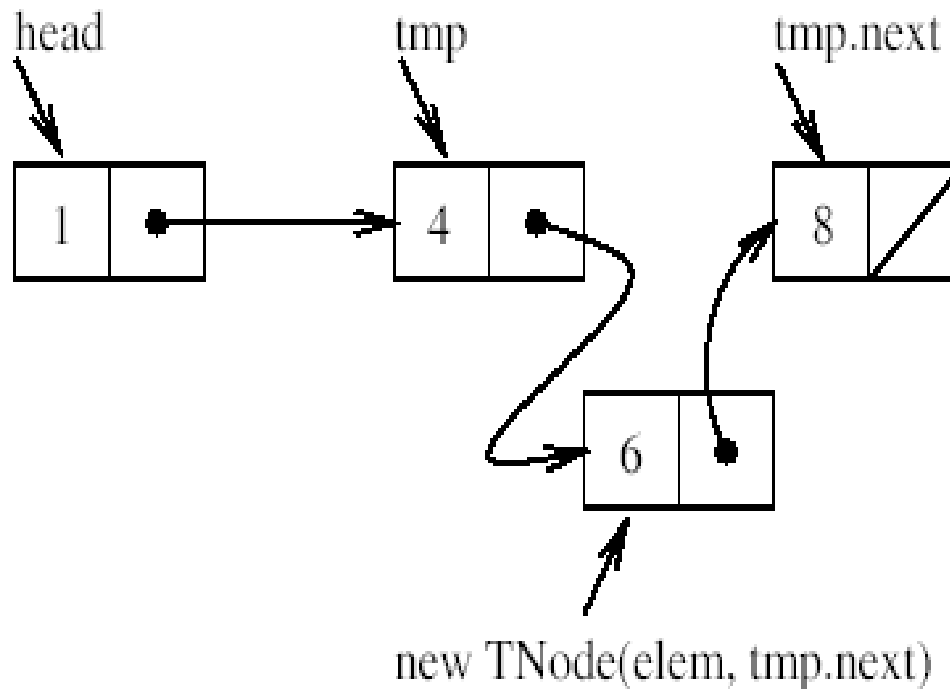


Typische Operationen auf Listen

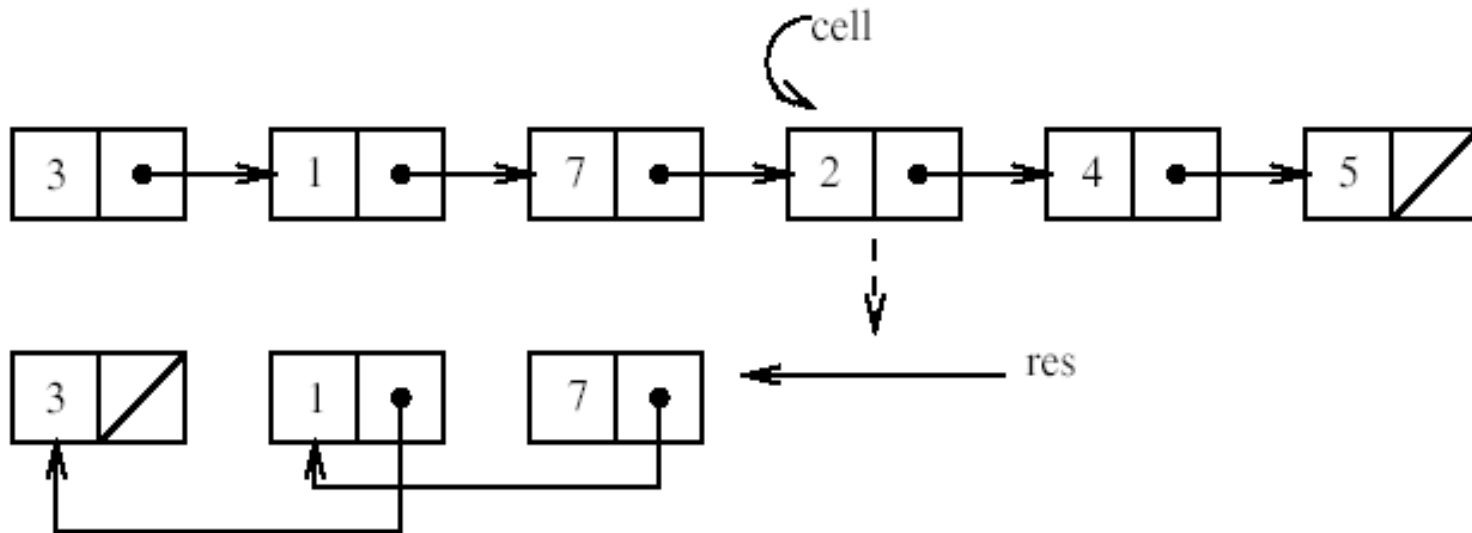
- Sortiertes Einfügen**



Wichtig:
Test auf
Sonderfälle

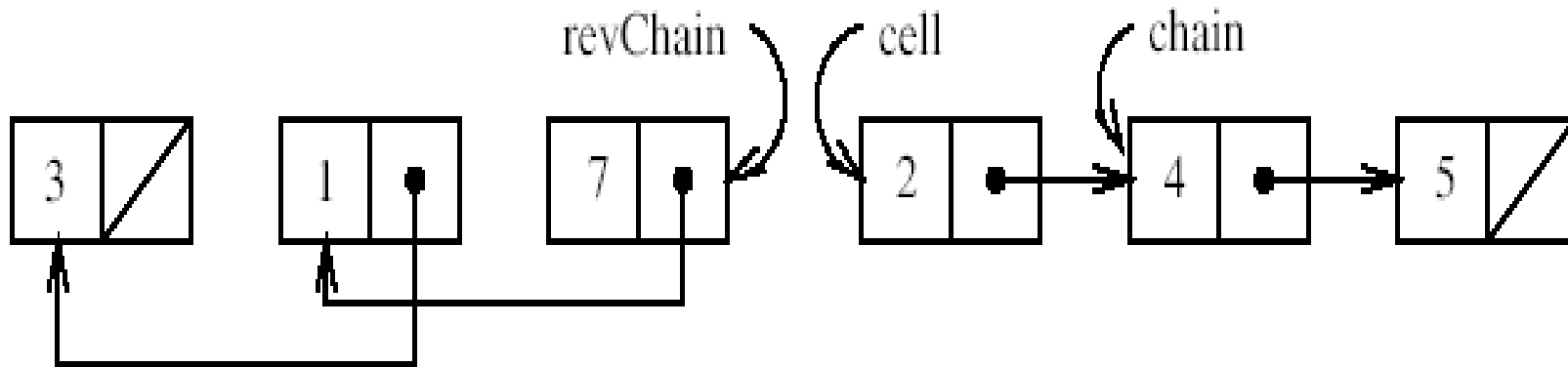
Typische Operationen auf Listen

- Konstruktives Invertieren**



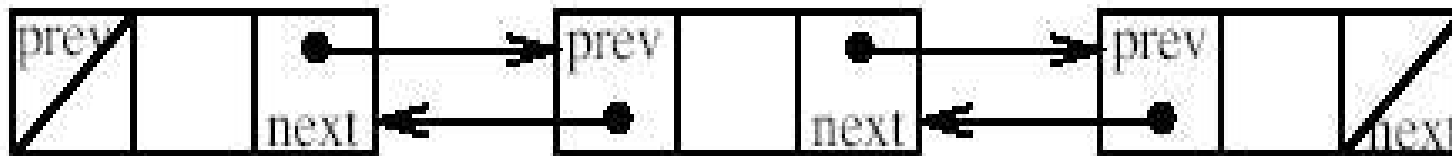
Typische Operationen auf Listen

- Destruktives Invertieren**



Doppelt verkettete Listen

- **Doppelt verkettete Listen** bestehen aus Listenzellen mit zwei Zeigern
 - ein Zeiger **prev** auf die vorherige Listenzelle
 - ein Zeiger **next** auf die nächste Listenzelle



Doppelt verkettete Listen

- **Vorteil** doppelt verketteter Listen
 - Einfügen am Ende sehr viel schneller möglich
Kein Durchlaufen durch die ganze Liste
- **Nachteile** doppelt verketteter Listen
 - Höherer Speicherbedarf
Zwei Referenzen pro Listenzelle statt einer
 - Mehr Aufwand bei Listenmanipulation
Zwei Referenzen sind zu ändern statt einer

Abstrakter Datentyp Stapel

- Stapel (*Keller, Stack*)
 - Datenstrukturen zum Zwischenspeichern von Elementen
 - Last-in-First-out-Prinzip (*lifo*)
 - Die Elemente, die als *letzte* auf dem Stapel abgelegt wurden (*push*), werden als *erste* zurückgegeben (*pop*)

Abstrakter Datentyp Stapel

- Implementierung von Stacks
 - Darstellung als **Containerklasse**, die
 - eine **einfache Liste** zur **Speicherung** der Elemente enthält
 - neben Konstruktoren bloß Methoden enthält, welche auf das erste Element zugreifen
 - **pop** für leeren Stack nicht definiert (partielle Funktion), dann Exception

Warteschlangen (Queues)

- Warteschlange (queue)
- Duales Konzept zum Stapel
- First-in/First-out-Prinzip (fifo)
- Die Elemente, die als **erste** in eine Warteschlange eingereiht wurden, werden auch als **erste** zurückgegeben.