

# Globalübung DSAL

mit Henri, Stefan und Jan

24. April 2019

# Organisatorisches

- ▶ Sind Räume immer noch zu voll?

# Rückblick auf Blatt 01

## Blatt 01, H2:

Implementieren Sie eine neue Methode *reverse()* in der Klasse *ADList*, welche die Reihenfolge in der Liste umkehrt. Dabei sollten Sie so vorgehen, dass kein zusätzlicher Speicher verwendet wird (abgesehen von konstant vielen Variablen), die Implementierung möglichst effizient ist und keine Rekursion verwendet wird. [...]

# Rückblick auf Blatt 01

## Blatt 01, H2:

Implementieren Sie eine neue Methode *reverse()* in der Klasse *ADList*, welche die Reihenfolge in der Liste umkehrt. Dabei sollten Sie so vorgehen, dass kein zusätzlicher Speicher verwendet wird (abgesehen von konstant vielen Variablen), die Implementierung möglichst effizient ist und keine Rekursion verwendet wird. [...]

- ▶ Wieso der Aufwand?

# Ablauf

- ▶ Fragen im Vorfeld an  
tcs-teaching@informatik.rwth-aachen.de,  
Betreff GÜFRAGE
- ▶ Fragen aus dem Publikum
- ▶ Ergänzend und Wiederholend zu Vorlesung und  
Übungsbetrieb

# Optimale Suchbäume

Konstruieren Sie einen optimalen Suchbaum (in lexikographischer Sortierung) für die Schlüssel *David*, *Bob*, *Carol* und *Alice*. Auf diese wird mit den Wahrscheinlichkeiten 0.4, 0.2, 0.15 und 0.25 zugegriffen.

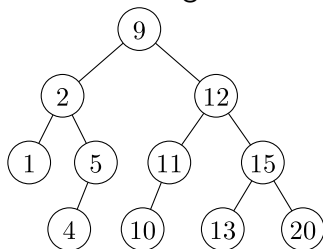
Erstellen Sie dazu die Tabellen für  $w_{i,j}$  und  $e_{i,j}$ .

# Optimale Suchbäume

- ▶ Ist jeder Unterbaum eines opt. Suchbaums selbst ein opt. Suchbaum?
- ▶ Was passiert bei nicht-lexikographischer Sortierung?

# Binärbäume: Rotationen

Rechts-& Linksrotationen auf folgendem Baum:



1. Rechtsrotation um 5
2. Rechtsrotation um 12
3. Rechtsrotation um 9
4. Linksrotation um 9



# AVL-Bäume: Konstruktion

Wir bauen einen AVL-Baum:

- ▶ Füge Knoten ein: 5, 3, 7, 9, 8, 6, 2, 1, 4

# AVL-Bäume: Konstruktion

Wir bauen einen AVL-Baum:

- ▶ Füge Knoten ein: 5, 3, 7, 9, 8, 6, 2, 1, 4

Wir machen den Baum ein bisschen kaputt:

- ▶ Lösche Knoten: 7, 9

# Knobelaufgabe: Sprouts