AVL-Baum T
- bin. Suchbaum
- Höhen der Kinder unterscheiden sich max. um 1

-> Suche in $O(\log(n))$ (n = #Knoten in T)
  da Höhe $\in O(\log(n))$

-> Einfügen, Suchen in $O(\log(n))$
  -> Aufrechterhaltung der AVL-Eigenschaft notw.
  -> restructure(x)

- korrekte Wahl von x und y, z:
  (Unklarheiten in letzter großen Übung)
  - Falls nach Löschen von Knoten v T nicht mehr AVL-Eigenschaft erfüllt (→ unbalancierter Knoten)
    - 1. Sei z niedrigster Knoten, der unb.
    - 2. " y Kind von z mit größerer Höhe
    - 3. " x " " y " " "

    äquivalent zu
    - 1. Sei z niedrigster Knoten, der unb.
    - 2. Sei y Kind von z, das nicht in Teilbaum, in dem v lag
    - 3. Sei x Kind von y mit größerer Höhe

    Äquivalenz, da nach Löschen von v unb.
    => Teilbaum in dem v lag hat kleinere Höhe

- Alternative Beschreibung gilt auch für Situation nach Einfügen, da
  - x, y, z werden auf Pfad zwischen z und v gewählt
  - v vergrößert Höhe des Teilbaumes

Sortieren:

(i) mit AVL-Bäumen

Array A, n Elemente

↓

AVL Baum aufbauen mit Werten aus A (Einfügen jeweils $O(\log n)$) — $O(n \log n)$

↓

AVL Baum in sortierter Reihenfolge auslesen + nach A schreiben — $O(n)$

$O(n \log n)$

Untere Schranke für Sortieren (allgemein): $\Omega(n \log n)$
[siehe VL ~~nächste Woche~~]

Warum noch Mergesort? Speicheraufwand!

mit AVL-Baum: $O(n)$ zusätzlicher Speicher

~~bei~~ Mergesort: $O(1)$ —″— —″—

(ii) Mergesort

(II)

A: 2 8 6 5

2 8 | 6 5

2 | 8 | 6 5

2 8 | 6 | 5

2 8 | 5 6

2 5 6 8

A[1]=2, A[2]=8, A[3]=6, A[4]=5

Mergesort (A,1,4)

$q = \lfloor \frac{5}{2} \rfloor = 2$

Mergesort (A,1,2)

$q = \lfloor \frac{3}{2} \rfloor = 1$

Mergesort(A,1,1) ← nichts passiert

Mergesort(A,2,2) ← ——— " ———

Merge (A,1,1,2) → 2 8 (*) ← nicht, sondern (**)

Mergesort (A,3,4)

$q = \lfloor \frac{7}{2} \rfloor = 3$

Mergesort (A,3,3) ← nichts passiert

Mergesort (A,4,4) ← nichts passiert

Merge(A,3,3,4) → 5 6 (**)

Merge (A,1,2,4) → 2 5 6 7 (***)

(*):

Merge(A, 1, 1, 2)

$\uparrow$ p, $\uparrow$ q, $\uparrow$ r

$n_1 = q - p + 1 = 1 - 1 + 1 = 1$

$n_2 = r - q = 2 - 1 = 1$

L[1,2]   R[1,2]

$i=1$: $L[1] = A[1+1-1] = A[1] = 2$

$j=1$: $R[1] = A[1+1] = A[2] = 8$

$L[2] = \infty$

$R[2] = \infty$

$i=1$

$j=1$

$k=1$: $L[1] \leq R[1]$? ($2 \leq 8$? ja)

$\hookrightarrow$ $A[1] = L[1] = 2$

$i = 2$

$k=2$: $L[2] \not\leq R[1]$ ($\infty \not\leq 8$)

$\hookrightarrow$ $A[2] = R[1] = 8$

end

Somit: $A[1] = 2$, $A[2] = 8$

(**)

Merge (A, 3, 3, 4)

$\uparrow$ p, $\uparrow$ q, $\uparrow$ r

$n_1 = q - p + 1 = 3 - 3 + 1 = 1$

$n_2 = r - q = 4 - 3 = 1$

L[1,2], R[1,2]

$i=1$: $L[1] = A[3 + 1 - 1] = A[3] = 6$

$j=1$: $R[1] = A[3+1] = A[4] = 5$

[7, 1, 8, 2, 5, 3, 4]

[7, 1, 8, 2] [5, 3, 4]

[7, 1] [8, 2] [5, 3, 4]

[7] [1] [8, 2] [5, 3, 4]

[1, 7] [8, 2] [5, 3, 4]

[1, 7] [8] [2] [5, 3, 4]

[1, 7] [2, 8] [5, 3, 4]

[1, 2, 7, 8] [5, 3, 4]

[1, 2, 7, 8] [5, 3] [4]

[1, 2, 7, 8] [5] [3] [4]

[1, 2, 7, 8] [3, 5] [4]

[1, 2, 7, 8] [3, 4, 5]

[1, 2, 3, 4, 5, 7, 8]