DSAL

1. Globalübung

Sebastian Junges, David Korzeniewski

24. April 2018

Agenda

1. Organisatorisches
2. Relationen
3. Pseudocode
4. Laufzeitanalyse
5. Traversierung von Binärbäumen

Organisatorisches

VL - Neuer Stoff

GÜ - Erklärung mit Beispielen (Wenig neuer Stoff)

Tutorium - Lösung der korrigierten Aufgaben mit Erklärung

SÜ - Weitere Erklärung zu schwierigen Hausaufgaben

Was ist eine binäre Relation?

Grundmenge Y

R relation $\quad R \subseteq Y \times Y$

$aRb \quad R = \{ (a,b) \mid a,b \in Y,$ a steht in relation zu b $\}$

Bsp I: $Y = \mathbb{N} \quad \{ (a,b) \mid \exists k \geq 0 \;\; a+k = b \} = \leq$

$a \leq b$ → Partielle Ordnung (sogar total)

Bsp II: $Y =$ Binarbäumen $\quad R = \{(a,b) \mid$ höhe$(a) \leq$ höhe$(b)\}$

$=$ ⇝ Äquivalenz

Bsp III: $Y =$ DAG $= (V,E)$ "reachability"

$R = \{ (s,t) \mid \exists v_1 \ldots v_n \in V, \; v_1 = s, \; v_n = t$
$\forall i < n \;\; (v_i, v_{i+1}) \in E \}$

↓ Partielle Ordnung

Eigenschaften:

- Reflexiv: $\forall x \in Y \quad xRx$
- Symmetrisch: $\forall x,y \in Y \quad xRy \Leftrightarrow yRx$
- Anti-Sym. $\forall x,y \in Y \quad xRy \wedge yRx \Rightarrow x=y.$

Beachte:

Anti-Sym $\neq$ nicht sym.

nicht sym: $\neg(\forall x,y \in Y \quad xRy \Leftrightarrow yRx)$

$\exists x,y \in Y \quad \neg(xRy \Leftrightarrow yRx)$

- Transitivität $\forall x,y,z \in Y \quad xRy \wedge yRz \Rightarrow xRz$

Partielle Ordnung: reflexiv, anti-sym, transitiv

Äquivalenz: reflexiv, sym, transitiv

Äquivalenzrelation $\sim$

Äquivalenzklassen: $[a] = \{ b \in Y \mid a \sim b \}$

"lineare Komplexität" Laufzeit f, $f \in \Theta(n)$

"linear"

$$\sim_\Theta = \{ (f,g) \mid f,g : \mathbb{R} \to \mathbb{R},\ f \in \Theta(g) \}.$$

Frage: Zeigen Sie, dass $\sim_\Theta$ eine Äquivalenzrelation ist.

- $f \sim_\Theta f$ z.Z. $f \in \Theta(f)$

$\exists\, n_0, c_1, c_2 \geq 0$

$0 \leq \underline{c_1 f(n) \leq f(n)} \leq c_2 f(n) \; \forall n \geq n_0$

wählen also $n_0 = 0$, $c_1 = 1$, $c_2 = 1$. $\square$

- $f \sim_\Theta g \Leftrightarrow g \sim_\Theta f$.

"$\Rightarrow$" Angenommen $f \sim_\Theta g$

$\exists\, n_0, c_1, c_2 \geq 0 \quad 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$

$\forall n \geq n_0$

z.Z. $\exists\, n_0', c_1', c_2' \geq 0$

$0 \underset{①}{\leq} c_1' f(n) \underset{②}{\leq} \underbrace{g(n) \overset{③}{\leq} c_2' f(n)}$,

$g(n) \leq c_2' f(n)$

$f(n) \geq c_1 g(n)$

$g(n) \leq \frac{1}{c_1} f(n)$. Wählen c_2' als $= \frac{1}{c_1}$

① , ② analog , genau wie "$\Leftarrow$"

$\cdot\ \underbrace{f \sim_\Theta g \wedge g \sim_\Theta h \Rightarrow}\ f \sim_\Theta h$

Angenommen φ. φ

$\exists n_0^1, c_1^1, c_2^1 \geq 0 \quad \ldots g(n) \leq f(n) \leq g(n) \ldots$

$\exists n_0^2, c_1^2, c_2^2 \geq 0 \quad \ldots h(n) \leq g(n) \leq \ldots h(n).$

zu zeigen: $\exists n_0, c_1, c_2 \geq 0 \quad 0 \leq c_1 h(n) \leq f(n) \leq c_2 h(n)$

$f(n) \leq c_2^1 g(n) \quad \forall n \geq n_0^1$

auch für $\forall n \geq \max(n_0^1, n_0^2)$

$g(n) \leq c_2^2 h(n) \quad \forall n \geq \max(n_0^1, n_0^2)$

positiv.

$\Rightarrow c_2^1 g(n) \leq c_2^1 c_2^2 h(n)$

$f(n) \leq c_2 h(n) \quad$ mit $c_2 = c_2^1 c_2^2$.

positiv.

Pseudocode

Pseudocode

KEIN Pseudocode

```
import java.util.Vector
public class MaximumFinder {
  public Integer findMaximum(Vector<Integer> arr) {
    Integer max = Integer.valueOf(Integer.MIN_VALUE);
    for(Integer val : arr) {
      if(val > max) {
        max = val;
      }
    }
    return max;
  }
}
```

Pseudocode

Pseudocode

```
Input: Ein Array von Integern
Output: Ein Integer

findMaximum(arr) {
  max = −∞
  for each (val in arr) {
    if (val > max) max = val
  }
  return max
}
```

Pseudocode

Pseudocode

```
findMaximum(arr: Array(Int)) : Int
  max : Int := minimal Int value
  for i := 0 .. arr.length -1
    if arr[i] > max
      max := arr[i]
  return max
```

Pseudocode

Pseudocode

```
DFS(G)
for each vertex u ∈ G.V
  u.color = WHITE
  u.π = nil
time = 0
for each vertex u ∈ G.V
  if u.color == WHITE
    DFS-Visti(G, u)
```

Pseudocode

Auch **kein** Pseudocode

Laufe über das Array und gib das Element zurück, das größer als jedes andere ist.

Laufzeitanalyse

DANKE !

der längen

```
Eingabe: Liste l von Zahlen zwischen 0 und k
Ausgabe: Gibt es Duplikate von Zahlen in l

gesehen = false^k+1.
for (i in l)
  if gesehen[i]
    return true
  else
    gesehen[i] = true

return false
```

[false, false, ...].

Best = [1, 1, ...]

$B(n,k) = 2$

Worst = [1, 2, 3, ...]

$W(n,k) = min(n,k)$

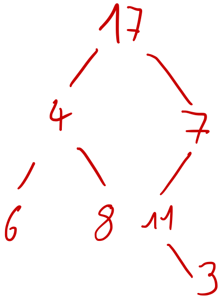
Average case : $k = 3$ (4 verschiedene Zahlen)

Jede Zahl ist gleichwahrscheinlich.

Iterationen	w'keit duplikat.	terminiere nach genau i Iterationen
1	0	0
2	$\frac{1}{4}$	$\frac{1}{4}$
3	$\frac{2}{4}$	$\frac{1}{2}\left(1-\frac{1}{4}\right) = \frac{3}{8}$
4	$3/4$	$\frac{3}{4}\left(1-\frac{1}{4}-\frac{3}{8}\right) = \frac{9}{32}$
5	1	$1\left(1-\frac{1}{4}-\frac{3}{8}-\frac{9}{32}\right) = \frac{3}{32}$
$\vdots$	1	

$$A(n) = 2 \cdot \frac{1}{4} + 3 \cdot \frac{3}{8} + 4 \cdot \frac{9}{32} + 5 \cdot \frac{3}{32} = c.$$

Traviersierung



inorder:

inorder(left)
print(val)
inorder(right)

6 4 8 17 11 3 7

postorder

post(left)
post(right)
print(val)

6 8 4 3 11 7 17

preorder:

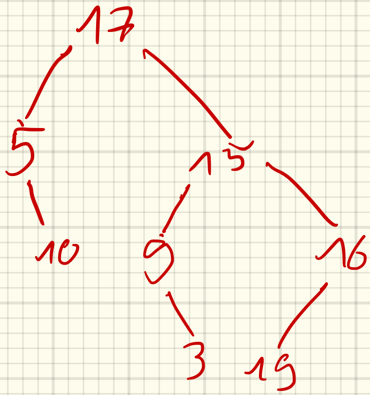
print(val)
pre(left)
pre(right)

17 4 6 8 7 11 3

Inorder: 5 10 17 9 3 15 19 16

Postorder: 10 5 3 9 19 16 15 17

L	L
V	R
R	V



Nächster Termin

Nächste Vorlesung

Freitag 27. April, 13:15 (H01).

Nächste Globalübung

Freitag 4. Mai, 13:15 (H01). (Statt Vorlesung)