13:45 Partnerbörse mit Martino

Kap 8 Rekursion

Aufruf ≙ Regelaktion

Selbstaufruf möglich "Rekursion" (direkt)

Beispiele: ggT

Türme von Hanoi

gelöst wie folgt

→ Schicke n-1 Scheiben von 1 nach 3

oberste Scheibe von 1 nach 2 ← nur dies "selbst gemacht"

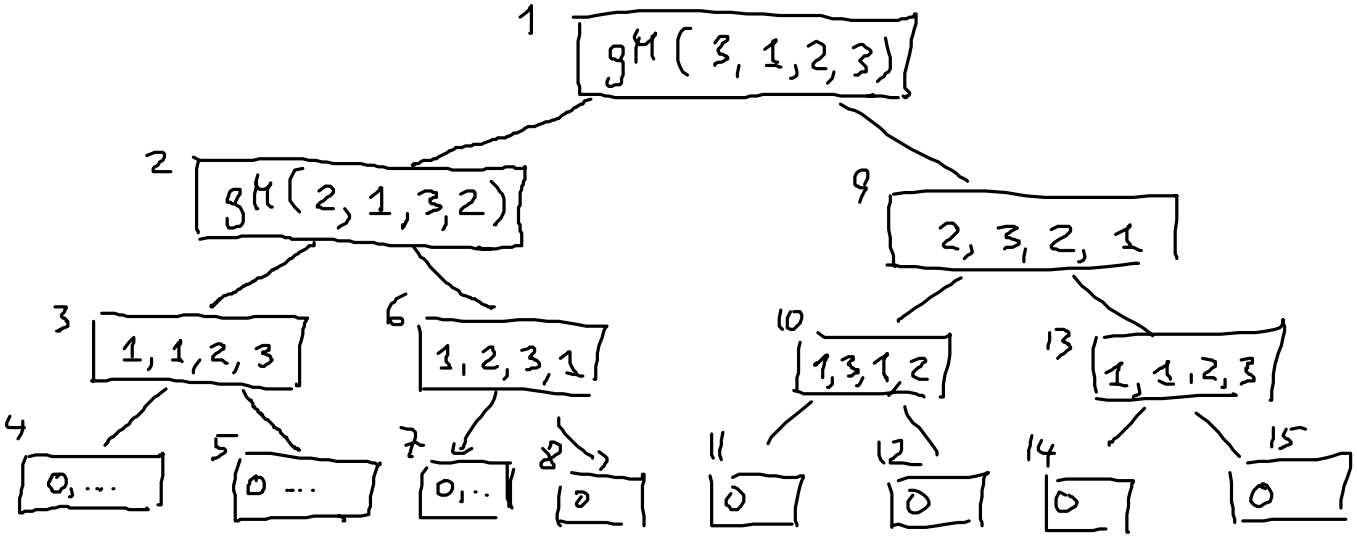
→ Schicke n-1 Scheiben von 3 nach 2

delegiert

Wichtig bei Rekursion

- Rekursionstiefe ← Maß für Speicher
- # rekursive Aufrufe ← Maß für Laufzeit

Beispiel getMoves (3, 1, 2, 3)



1 →→ 2
1 →→ 3
2 →→ 3
1 →→ 2
3 →→ 1
3 →→ 2
1 →→ 2

↑ Rekursionsbaum

Rekursionstiefe = 4

rek Aufrufe = 15

Run-Time-Stack

1	2	3	4	3	5	...
	1	2	3	2	3	
		1	2	1	2	
			1		1	

↑ Rek. Tiefe 4

Korrektheitsbeweis mit Induktion und # Schritten

Rekursion & Induktion gehen Hand in Hand

SATZ: $R(n)$:= Rekursionstiefe bei n Schritten

$T(n)$:= # rek. Aufrufe bei n Schritten

Es gilt: $R(n) = n+1$

$T(n) = 2^{n+1} - 1$

$Z(n) :=$ Anzahl Zeige $\quad Z(n) = 2^n - 1$

Beweis mit Induktion

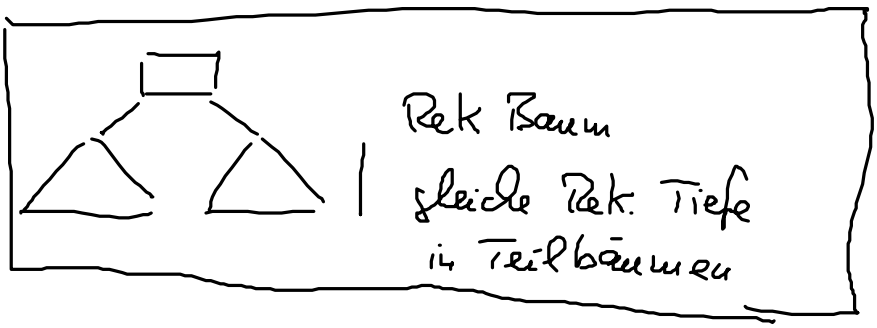
$n = 0 \quad \Rightarrow$ 1 Aufruf 0 Zeige ✓

IV. Beh sei richtig für $0, 1, 2, \dots, n-1$ Scheiben

Schluss auf n

$R(n) = 1 + R(n-1)$

$\overset{IV}{=} 1 + n$ ✓



$T(n) = 1 + 2 \cdot T(n-1)$

$\overset{IV}{=} 1 + 2 \cdot \left(2^{n-1+1} - 1\right) = 1 + 2^{n+1} - 2$

$= 2^{n+1} - 1$ ✓

$Z(n) = Z(n-1) + 1 + Z(n-1)$

$= 2^{n-1} - 1 + 1 + 2^{n-1} - 1 = 2^n - 1$ ✓

zu  : Rekursion bricht erst dann ab
wenn # Scheiben = 0 ist

⇒ im Rek. Baum sind alle Stichten
voll besetzt mit Knoten

⇒ linker Teilbaum und rechter Teilbaum
haben gleiche Knotenzahl und gleiche
Rek. tiefe □

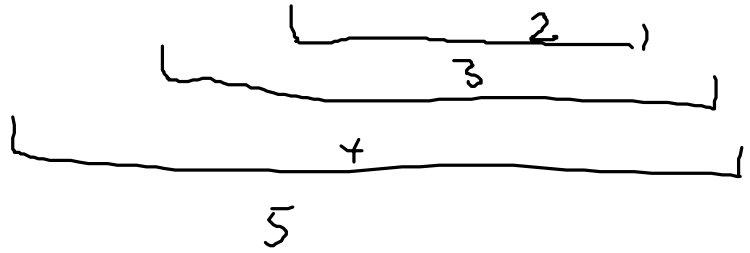
8.4.3. Die Ackermann Funktion

Beispiel einer sehr stark wachsenden
rekursiv def. Fkt

$$a(m,n) := \begin{cases} n+1 & \text{falls } m = 0 \\ a(m-1,1) & \text{falls } m > 0,\ n = 0 \\ a(m-1, a(m, n-1)) & \text{falls } m, n > 0 \end{cases}$$

Bsp:

$$\begin{aligned} a(1,3) &= a(0, a(1,2)) \\ &= a(0, a(0, a(1,1))) \\ &= a(0, a(0, a(0, a(1,0)))) \\ &= a(0, a(0, a(0, \underline{a(0,1)}))) \end{aligned}$$



$a(0,n) > n$

$a(1,n) > n+1$

$a(2,n) > 2n$

$a(3,n) > 2^{2^{\cdots^{2}}}\Big\}\, n$

$a(4,n) > 2^{2^{\cdots^{2}}}\Big\}\, n$

$a(5,n)$? keine geschlossene Formel, $a(5,4) > 10^{10.000}$

SATZ: Für alle m, n terminiert der Aufruf $a(m,n)$ nach endlich vielen Schritten

Beweis: vollst. Induktion (geschachtelte Induktion)

Induktion nach m (äußere Induktion)

Ind. Anf $m = 0$ $\Rightarrow$ terminiert für alle n

Ind. Anw: Aufruf terminiert für Werte $< m$ in erster Komp und für alle n

Schluss auf m

Zeige: Aufruf $a(m,n)$ terminiert für alle n
(m jetzt fest

Ind. nach n (innere Induktion)

Ind. Anf. $n = 0$

$$a(m,0) = \underbrace{a(m-1,1)}$$

terminiert nach äußerer Ind. Vor.

Ind. Vor.: $a(m,k)$ terminiert für $k = 0, 1, 2, \ldots, n-1$

Schluss auf n

$$a(m,n) \overset{\text{Def}}{=} a(m-1, \underbrace{a(m,n-1)})$$

terminiert nach innerer Ind. Vor.
und liefert eine Zahl N

$$= \underbrace{a(m-1,N)}$$

terminiert nach äußerer Ind. Ann. □

8.1.4 Ackermann Funktion

Beispiel für einfache rekursive Fkt,
von der unbekannt ist ob sie bei allen Aufrufen terminiert

Algorithmus erzeugt Folge von Zahlen

$$a_0 > 0 \;,\; a_1 \;,\; a_2 \;,\; a_3 \;\ldots\ldots \qquad a_0 \in \mathbb{N}$$

mit $$a_n := \begin{cases} a_{n-1}/2 & \text{falls } a_{n-1} \text{ gerade} \\ 3 \cdot a_{n-1} + 1 & \text{sonst} \end{cases}$$

und bricht ab, wenn $a_n = 1$ ist

$$a_0 = 3 \longrightarrow 10 \longrightarrow 5 \longrightarrow 16 \longrightarrow 8 \longrightarrow 4 \longrightarrow 2 \longrightarrow 1$$
$$a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7$$

0

Ulams Funktion

$$ulam(a) := \begin{cases} 0 & a = 1 \\ 1 + ulam(a/2) & a \text{ gerade} \qquad a \in \mathbb{N} \\ 1 + ulam(3a+1) & a \text{ ungerade} \qquad a \in \mathbb{N} \end{cases}$$

$ulam(3) = 7$

ulam(a) zählt Länge der Folge bis man bei 1 ist

8.2 Wo Rekursion zu vermeiden ist

Abwägung: Rekursion oder Iteration

schlecht, wenn ↙ ↘ gut, wenn Problem

Run-Time-Stack
$\hat{=}$ Rek. Tiefe
zu groß wird

eine rekursive Struktur hat
(Türme von Hanoi)

8.2.1 Fakultät

$$fak(n) = \begin{cases} 1 & n = 0 \\ n \; fak(n-1) & n > 0 \end{cases} \qquad n \in \mathbb{N}$$

Rekursionstiefe: $n+1$ ← Stack enthält $n-1$ Kopien des Parameters n

einfache iterative Lösung: $n! = n(n-1)(n-2)\ldots 1 \quad n \geq 1$

rekursiv

```
if (n == 0) fak = 1;
else
 fak = n * fak(n-1);
```

einfach mit Iteration

```
fak = 1;
for ( k = 2; k <= n; k++) {
        fak *= k;
}
```

8.2.2. Berechnung ggT

gegen Rekursion: Rek. Baum ist Liste

dafür Rek. Tiefe klein $1 + 2\log(\max\{a,b\})$

in Summe: okay

8.2.3. Türme von Hanoi

Rek. Baum ist keine Liste ⇒ Iteration möglicherweise schwierig zu realisieren

Rek. Tiefe, # rek. Aufrufe sind groß

$n+1$ $2^{n+1}-1$ ⇐ schlecht

Struktur des Problems ist rekursiv ⇐ gut

allerdings ist $Z(n) = 2^n - 1$
und jeder Zug muss berechnet werden

in Summe: okay

8.2.4. Berechnung der Fibonacci Zahlen

$f_0, f_1, f_2, \ldots$ Zahlenfolge mit Bildungsgesetz

$f_0 := 0, \quad f_1 := 1$

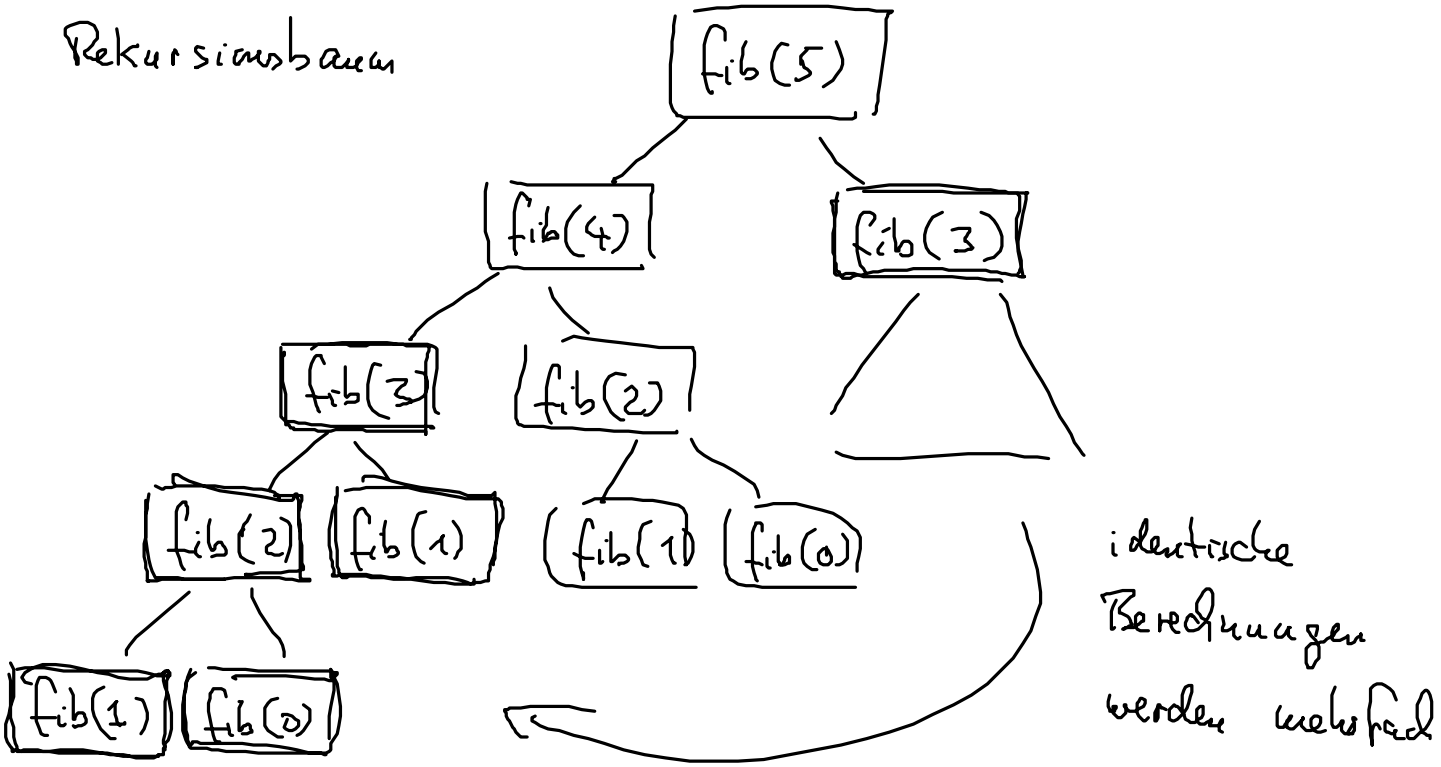
$f_{n+1} := f_n + f_{n-1}$

0 1 1 2 3 5 8 13 21 34 ....

Methode zur Berechnung der n-ten Fib. Zahl

```
int fib ( int n ) {
    if ( n == 0 ) return 0;
    else if ( n == 1 ) return 1;
    else return fib(n-1) + fib(n-2);
}
```

Rekursionsbaum



identische Berechnungen werden mehrfach angestellt

nicht sinnvoll

Satz: $\boxed{T_{fib}(n) \geq 2^{\lfloor n/2 \rfloor}}$

↑

rek Aufrufe wächst exponentiell

Beweis: $T_{fib}(n+1) \geq 2 \cdot T_{fib}(n-1)$

$\overset{IV}{\geq} 2 \cdot 2^{\lfloor n-1/2 \rfloor} = 2 \cdot 2^{\lfloor (n+1)/2 \rfloor}$

Es gibt einfache iterative Lösung (Skript)