AuD: große Übung
30.10.2014

Christian Scheffer

Algorithmus

Lösung

Problem

Algorithmus

Lösung

Problem

Beispiel: Fibonacci-Zahlen

Problem ⇨ Algorithmus ⇨ Lösung

Berechne f(5)

5

Probleminstanz

$$
\begin{aligned}
f(5) &= f(4) + f(3) \\
&= f(3) + f(2) + f(2) + f(1) \\
&= f(2) + f(1) + f(2) + f(2) + f(1) \\
&= 1 + 1 + 1 + 1 + 1 = 5
\end{aligned}
$$

$$f(1) = 1,\ f(2) = 1,\ f(n) = f(n-1) + f(n-2) \text{ für } n > 2$$

Beispiel: Fibonacci-Zahlen

Problem ⇨ Algorithmus ⇨ Lösung

Berechne f(5) — 5

Probleminstanz

Algorithmus zur Berechnung von f(5):

Gib 5 zurück

$$f(1) = 1,\ f(2) = 1,\ f(n) = f(n-1) + f(n-2) \text{ für } n > 2$$

5

Beispiel: Fibonacci-Zahlen

Problem ⇨ Algorithmus ⇨ Lösung

Berechne eine Fibonacci-Zahl

bel. Fib.-Zahl

Algorithmus zur Berechnung bel. Fib. Zahl

~~Gib 5 oder 1 aus~~

Gib 5 aus

Eindeutigkeit

$$f(1) = 1,\ f(2) = 1,\ f(n) = f(n-1) + f(n-2) \text{ für } n > 2$$

Beispiel: Fibonacci-Zahlen

Problem ⇨ Algorithmus ⇨ Lösung

Berechne n-te Fibonacci-Zahl

n-te Fib.-Zahl

Algorithmus zur Berechn. der n-ten Fib.-Zahl

f(1)=1, f(2)=1, f(3)=2, f(4)=3, f(5)=5, f(6)=8, f(7)=13,
f(8)=21, f(9)=34, f(10)=55, f(11)=89, f(12)=144,
f(13)=233, f(14)=377 f(15)=610, f(16)=987, f(17)=1597
f(18)=2584, f(19)=4181, f(20)=6765, f(21)=10946
f(22)=17711, f(23)=28657, f(24)=46368, f(25)=75025,
f(26)=121393, f(27)=196418, f(28)=317811, ...

- ~~Gib n-te Fib. Zahl aus~~

Eindeutigkeit

Endlich

Laufzeit beschreibbar

$$f(1) = 1,\ f(2) = 1,\ f(n) = f(n-1) + f(n-2) \text{ für } n > 2$$

Beispiel: Fibonacci-Zahlen

Problem ⇨ Algorithmus ⇨ Lösung

Berechne n-te Fibonacci-Zahl

n-te Fib.-Zahl

Algorithmus zur Berechn. der n-ten Fib.-Zahl

- f(1)=1
- f(2)=1
- f(3)=f(2)+f(1)=2
- f(4)=f(3)+f(2)=2+1=3
- f(5)=f(4)+f(3)=3+2=5

...

- f(n)=f(n-1)+f(n-2)=...

n Schritte

Eindeutigkeit

Endlich bzgl. Eingabe (n)

Laufzeit beschreibbar

$$f(1) = 1,\ f(2) = 1,\ f(n) = f(n-1) + f(n-2) \text{ für } n > 2$$

Problem ⇨ Algorithmus ⇨ Lösung

ggT(a,b) | euklidischer Algorithmus | ggT(a,b)

Euklidischer Algorithmus für zwei Zahlen:

falls a gleich 0, gib b zurück

falls a ungleich 0, wiederhole Folgendes
solange b ungleich 0 ist

falls a größer als b, verringere a um b
falls a kleiner oder gleich b,
verringere b um a

gib a zurück

Problem ⇨ Algorithmus ⇨ Lösung

ggT(a,b)

euklidischer Algorithmus

ggT(a,b)

```
int EuklAlgo(int a, int b){
    if(a==0){
        return b;
    }else{
        while(b!=0){
            if(a>b){
                a=a-b;
            }else{
                b=b-a;
            }
        }
    }
}
```

Problem ⇨ Algorithmus ⇨ Lösung

ggT(a,b)

euklidischer Algorithmus

ggT(a,b)

```
EuklAlgo(a,b)
    wenn a=b
    |   return b
    sonst
    |   solange b≠0
    |   |   falls a>b
    |   |    | a=a-b;
    |   |   sonst
    |   |    | b=b-a;
```

Problem ⇨ Algorithmus ⇨ Lösung

ggT(a,b)

euklidischer Algorithmus

ggT(a,b)

```
EuklAlgo(a,b)
1--wenn a=b
2----return b
3--sonst
4----so lange b≠0
5--------falls a>b
6----------a=a-b;
7--------sonst
8----------b=b-a;
```

a=32, b=8

a=24, b=8

a=16, b=8

a=8, b=8

a=8, b=0

Beispiel: Fibonacci-Zahlen

Problem ⇨ Algorithmus ⇨ Lösung

Berechne n-te Fibonacci-Zahl

n-te Fib.-Zahl

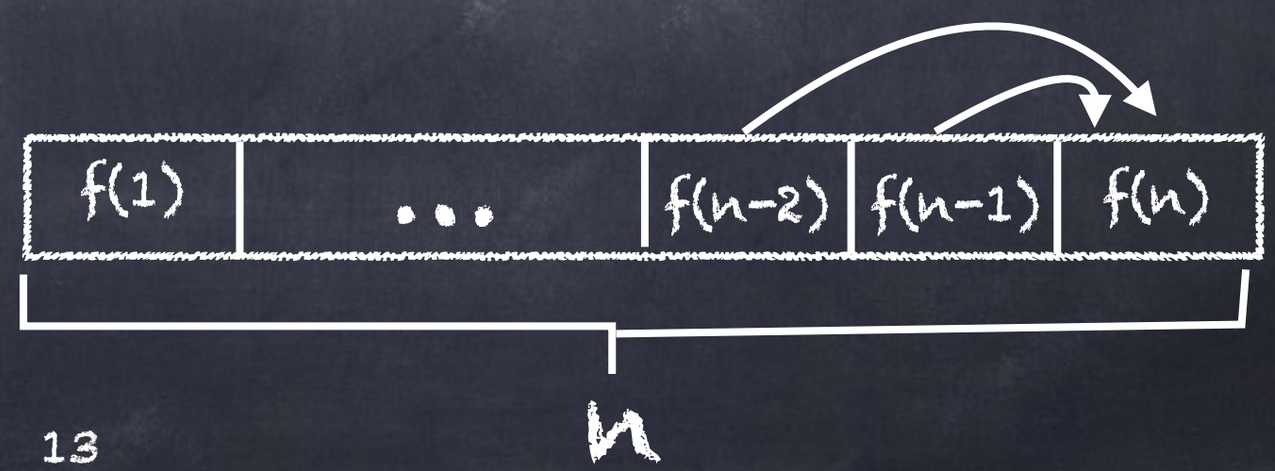
Algorithmus zur Berechn. der n-ten Fib.-Zahl

- $f(1)=1$
- $f(2)=1$
- $f(3)=f(2)+f(1)=2$
- $f(4)=f(3)+f(2)=2+1=3$
- $f(5)=f(4)+f(3)=3+2=5$

...

- $f(n)=f(n-1)+f(n-2)=\ldots$

- Merken von Ergebnissen
- Pro Iteration eine Zahl
- „lineare Komplexität"



Beispiel: Fibonacci-Zahlen

Problem ⇨ Algorithmus ⇨ Lösung

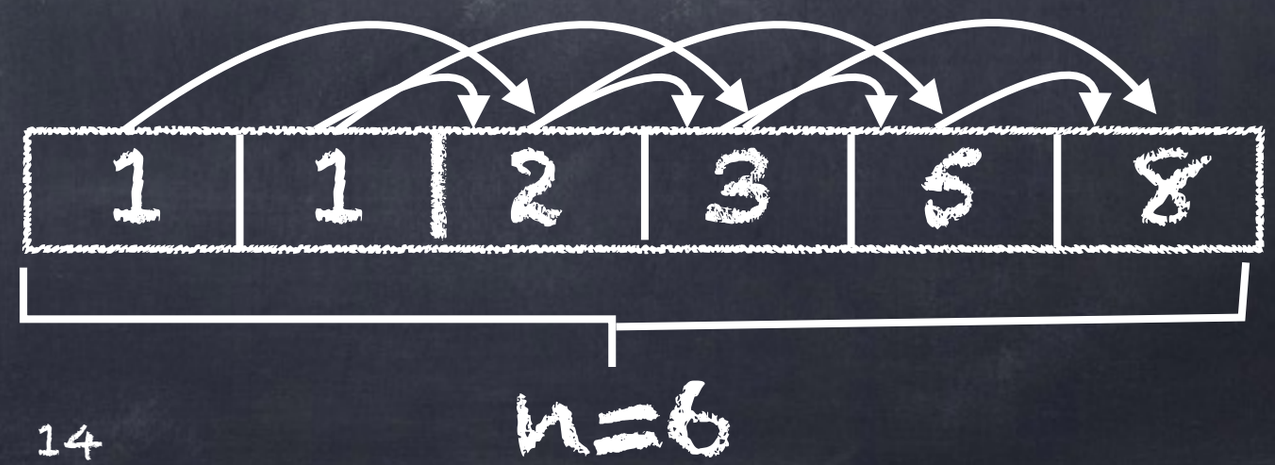
Berechne n-te Fibonacci-Zahl

n-te Fib.-Zahl

Algorithmus zur Berechn. der n-ten Fib.-Zahl

```
fib(n)
A=<0,...,0>
z=3
A[1]=1
A[2]=1
so lange z<=n
    A[z]=A[z-1]+A[z-2]
    z=z+1
return A[z-1]
```

- Merken von Ergebnissen
- Pro Iteration eine Zahl
- „lineare Komplexität"



Beispiel: Fibonacci-Zahlen

Problem ⇨ Algorithmus ⇨ Lösung

Berechne n-te Fibonacci-Zahl

n-te Fib.-Zahl

Algorithmus zur Berechn. der n-ten Fib.-Zahl

```
fib(n)
1- - - A=<0,...,0>
2- - - z=3
3- - - A[1]=1
4- - -A[2]=1
5- - - so lange z<=n
6- - - -  A[z]=A[z-1]+A[z-2]
7 - - - - z=z+1
8- - -return A[z-1]
```

Algorithmus zur Berechnung der n-ten Fib.-Zahl
Initialisiere Array der Größe n.
Setze initial globalen Zähler auf eins.
Setze initial ersten Wert des Arrays auf eins.
Setze initial den zweiten Wert des Arrays auf zwei.
Wiederhole die folgenden zwei Schritte, bis
der n-te Wert des Arrays berechnet ist.
Gib den n-ten Wert des Arrays zurück.

Beispiel: Fibonacci-Zahlen

Problem ⇨ Algorithmus ⇨ Lösung

Berechne n-te Fibonacci-Zahl

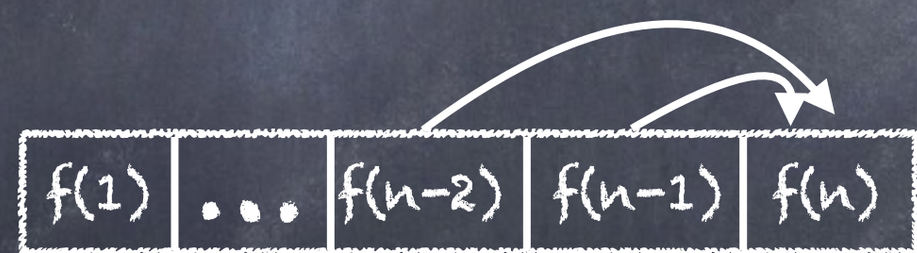
n-te Fib.-Zahl

Algorithmus zur Berechn. der n-ten Fib.-Zahl

Entwurf

$f(1)=1$
$f(2)=1$
$f(n)=F(n-2)+f(n-1)$

⇨



Skizze

⇨

```
fib(n)
1- - - A<0,...,0>
2- - - z=1
3- - - A[1]=1
4- - - A[2]=1
5- - - so lange z<=n
6- - - - - A[z]=A[z-2]+A[z-1]
7- - - - - z=z+1
8- - - return A[z]
```

(form.) Beschreibung

⇦

„lineare Komplexität"

Analyse

⇦

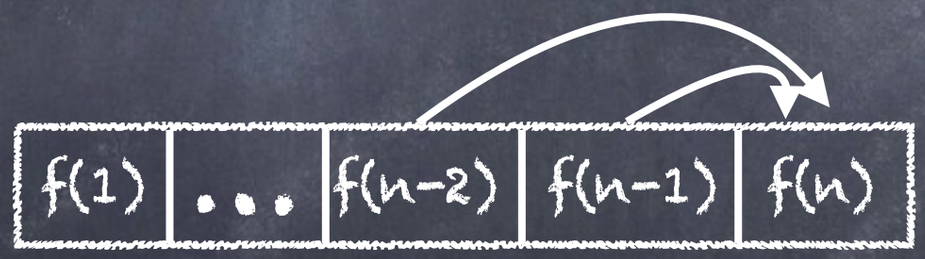
Pseudo-Code

Beispiel: Fibonacci-Zahlen

Problem → Algorithmus → Lösung

Algorithm Engineering

$f(1)=1$
$f(2)=1$
$f(n)=F(n-2)+f(n-1)$



Skizze

```
fib(n)
A<0,...,0>
z=1
A[1]=1
A[2]=1
so lange z<=n
    A[z]=A[z-2]+A[z-1]
    z=z+1
return A[z]
```

Pseudo-Code

„lineare Komplexität"

Analyse

Experiment

Implementierung

Zusammenfassung

- Problem, Algorithmus und Lösung
- Bsp.: Fibonacci-Zahlen, ggt(.,.)
- Eigenschaften von Algorithmen
  - Eindeutigkeit
  - Endlichkeit
- Pseudo-Code
- Entwurf von Algorithmen
- Analyse von Algorithmen

Fragen?