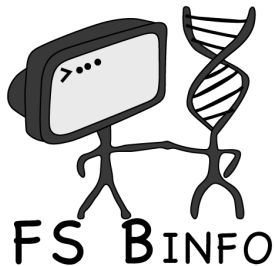


Rekursion vs. Iteration



Jeany Prinz

Fachschaft Bioinformatik

1. Dezember 2009

Zeitplan

- 27.10 Einführung in Linux und die Shell
 - 03.11 Arbeiten mit Dateien auf der Shell
 - 10.11 Einführung in Java
 - 17.11 Umgang mit Arrays, Schleifen und Dateien
 - 24.11 Einführung in objekt-orientierte Programmierung
 - 01.12 **Rekursion und Iteration**
 - 08.12 GUI Entwicklung
 - 15.12 Debuggen mit Eclipse
-
- 12.01 Taschenrechner in Java I
 - 19.01 Taschenrechner in Java II
 - 26.01 Wiederholung

Webseite: <http://www.bioinformatik-muenchen.com/bioinfocom/informatik-tutorium>

Rekursion

- lat. recurrere zurücklaufen
- Funktion wird durch sich selbst definiert
- Beispiel 1: Summe von 1 bis n

$$\text{sum}(n) = \begin{cases} 0 & \text{falls } n = 0 \\ \text{sum}(n - 1) + n & \text{sonst} \end{cases}$$

Rekursionsanfang
Rekursionsschritt

- Beispiel 2: grösster gemeinsamer Teiler

$$\text{ggT}(a, b) = \begin{cases} a & \text{falls } b = 0 \\ \text{ggT}(b, a \% b) & \text{sonst} \end{cases}$$

Rekursionsanfang
Rekursionsschritt

Iteration

- lat. wiederholen
- Wiederholung durch Schleifen, kein rekursiver Aufruf

- ggT iterativ:

```
while (b>0){  
    int r=a % b;  
    a=b;  
    b=r;}  
return a
```



Übung Fakultät

- Fakultät(n) ist als das Produkt der natürlichen Zahlen von 1 bis n definiert
- $\prod_{i=1}^n i = 1 \cdot 2 \cdot \dots \cdot n$
- Aufgabe: Implementiere die Fakultät sowohl iterativ als auch rekursiv

Rekursion vs. Iteration

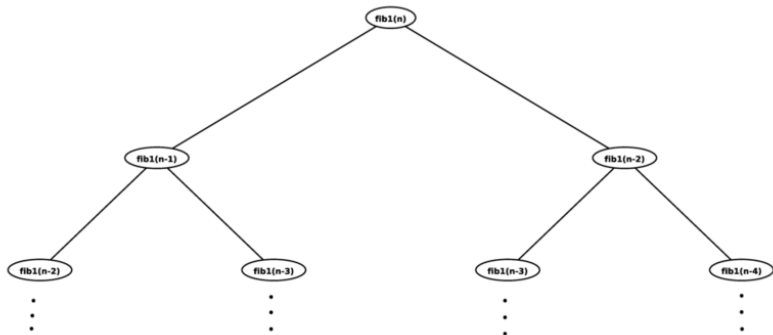
- bei Rekursion wird Programm immer wieder aufgerufen
⇒ Iteration ist meist effizienter
- Rekursive Lösungen legen u.a. die Werte der aktuellen Parameter und der lokalen Variablen auf dem Stack ⇒ benötigen mehr Arbeitsspeicher
- Rekursion ist intuitiver und meist übersichtlicher

Fibonacci

- Folge von Zahlen (den Fibonacci-Zahlen), bei der sich die jeweils folgende Zahl durch Addition der beiden vorherigen Zahlen ergibt:
- 0, 1, 1, 2, 3, 5, 8, 13, ...
- Rekursive Definition

$$Fib(n) = \begin{cases} n & \text{falls } n = 0 \text{ oder } n = 1 \\ Fib(n-1) + Fib(n-2) & \text{sonst} \end{cases} \quad \begin{array}{l} \text{Rekursionsanfang} \\ \text{Rekursionsschritt} \end{array}$$

Aufrufbaum der rekursiven Fibonaccifunktion



- Implementiere die Fibonaccifunktion (Rückgabe: n-te Fibonaccizahl) rekursiv und iterativ

Übung für zu Hause

- Schreibe ein rekursives und ein iteratives Programm das testet ob ein Wort ein Palindrom ist