



# Grundzüge der Informatik 1

Vorlesung 2 – flipped classroom

# Überblick

## Überblick

- Rekursion
- Speicheraufbau und Datentypen
- Pseudocodebefehle und Laufzeit
- Laufzeitanalyse

# Rekursion

Rückwärtszähler(n)

1. **output** << n
2. **if** n=0 **then return**
3. Rückwärtszähler(n-1)

## Was ist Rekursion?

- Rekursion bezeichnet den Selbstaufwurf von Algorithmen

# Rekursion

## Rekursion in der Algorithmenentwicklung

- Rekursion führt die Lösung eines Problems auf die Lösung eines einfacheren (typischerweise kleineren) Problems zurück
- Es gibt ein grundlegendes Problem, dass einfach ohne Rekursion gelöst werden kann (Rekursionsabbruch)

# Entwicklung eines Sortieralgorithmus

OurSort(A, n)

A ist Feld (Array) mit n Zahlen

1. **if**  $n=1$  **then return**
2.  $x=A[n]$
2. OurSort(A,n-1)
3. Füge x an die korrekte Stelle in A ein

# Rekursion

## Aufgabe 1

- Entwickeln Sie einen rekursiven Algorithmus, um  $n!$  (für  $n \geq 1$ ) auszurechnen
- Erinnerung:  $n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1$

# Rekursion

## Aufgabe 1

- Entwickeln Sie einen rekursiven Algorithmus, um  $n!$  (für  $n \geq 1$ ) auszurechnen
- Erinnerung:  $n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1$

Fakultät( $n$ )

1. **if**  $n=1$  **return** 1
2. **else**
3.     **return**  $n \cdot \text{Fakultät}(n-1)$

# Speicheraufbau und Datentypen

## Elementare Datentypen

- ganze Zahlen
- reellwertige Zahlen
- Zeichen
- Zeiger

## Speicher

- Beliebig viele Speicherzellen
- Speicherzellen sind mit 1 beginnend aufsteigend durchnummeriert
- Elementare Datentypen benötigen eine Speicherzelle



# Speicheraufbau und Datentypen

## Überblick

- Elementare Datentypen belegen eine Speicherzelle
- Der Speicherbedarf eines Feldes oder Verbundes entspricht der Anzahl seiner Elemente
- Neue Felder oder Verbundobjekte werden mit **new** erzeugt
- Der Speicherbedarf eines Algorithmus ist die Summe der belegten Speicherzellen und kann von der Eingabe abhängen

# Speicheraufbau und Datentypen

## Überblick

- Elementare Datentypen belegen eine Speicherzelle
- Der Speicherbedarf eines Feldes oder Verbundes entspricht der Anzahl seiner Elemente
- Neue Felder oder Verbundobjekte werden mit **new** erzeugt
- Der Speicherbedarf eines Algorithmus ist die Summe der belegten Speicherzellen und kann von der Eingabe abhängen

## Bemerkung

- Unser Rechenmodell abstrahiert von vielen Details moderner Hardware wie z.B. Speicherhierarchien

# Speicheraufbau und Datentypen

## Algorithmus1()

1. A = **new** array[1..100]
2. A[1] = 10
3. B=A
4. B[1] = 20
5. A[2] = 30
6. x = A[1]\*B[2]

## Aufgabe 2

- Was ist der Speicherbedarf des Algorithmus?
- Was ist der Wert der Variable x am Ende des Algorithmus?

# Speicheraufbau und Datentypen

## Algorithmus1()

1. A = **new** array[1..100]
2. A[1] = 10
3. B=A
4. B[1] = 20
5. A[2] = 30
6. x = A[1]\*B[2]

## Aufgabe 2

- Was ist der Speicherbedarf des Algorithmus?
- Der Speicherbedarf ist 103 Speicherzellen
- Was ist der Wert der Variable x am Ende des Algorithmus?
- Der Wert von x ist 600

# Speicheraufbau und Datentypen

Einfach3()

1. `x = new list_item`
2. `number[x] = 10`

Verbund list\_item:

previous  
number  
next

## Verbunddaten

- Auf die einzelnen Elemente eines Verbundes BEISPIEL wird mit `<element_name> [BEISPIEL]` zugegriffen

# Speicheraufbau und Datentypen

Verbund list\_item:

previous  
number  
next

## Aufgabe 3

- Schreiben Sie einen Pseudocode, der rekursiv für eine ganzzahlige positive Eingabevariable  $n$  eine Liste mit den Zahlen 1 bis  $n$  erzeugt
- Beschreiben Sie, wie eine Liste mit 3 Elementen im Speicher abgelegt ist

# Speicheraufbau und Datentypen

Verbund list\_item:

previous  
number  
next

Liste(n)

1. **if**  $n=0$  **then return** NIL
2. **else**
3.    $L = \text{Liste}(n-1)$
4.    $x = \text{new list\_item}$
5.    $\text{previous}[x] = \text{NIL}$
6.    $\text{number}[x] = n$
7.    $\text{next}[x] = L$
8.   **if**  $L \neq \text{NIL}$  **then**  $L.\text{previous} = x$
9.   **return**  $x$

## Aufgabe 3

- Schreiben Sie einen Pseudocode, der rekursiv für eine ganzzahlige positive Eingabevariable  $n$  eine Liste mit den Zahlen 1 bis  $n$  erzeugt
- Beschreiben Sie, wie eine Liste mit 3 Elementen im Speicher abgelegt ist

