

# 1. Übungsblatt

zur Vorlesung

## Grundzüge der Informatik I

Abgabe über Ilias bis zum 12.4. 14:00 Uhr.  
Besprechung in Kalenderwoche 16.

### Aufgabe 1 Rekursion (4 Punkte)

Geben Sie in Pseudocode einen rekursiven Algorithmus an, welcher für zwei gegebene Zahlen  $n$  und  $a$  das Produkt  $\prod_{i=0}^n a^i$  berechnet.

### Aufgabe 2 Felder (4 + 2 Punkte)

- a) Gegeben sei ein Feld  $A$  mit  $n$  Zahlen  $a_1, a_2, \dots, a_n$ . Gesucht ist das Paar zweier benachbarter Elemente im Array mit maximaler Differenz.

Geben Sie in Pseudocode einen rekursiven Algorithmus an, der das Paar  $(i, i + 1)$  berechnet, das  $|a_i - a_{i+1}|$  maximiert.

- b) Geben Sie die Laufzeit ihres Algorithmus aus Teilaufgabe a) an.

### Aufgabe 3 (4 + 2 Punkte)

In den folgenden Teilaufgaben sind eine Problemstellung und ein rekursiver Algorithmus zum Lösen des Problems gegeben. Jedoch haben sich Fehler in die Algorithmen eingeschlichen. Finden und korrigieren Sie die Fehler, sodass die betroffenen Algorithmen ihr Problem richtig lösen. Verändern Sie dabei so wenig wie möglich an dem vorhandenen Pseudocode!

- a) Gegeben sei eine natürliche Zahl  $n$ . Der folgende Algorithmus soll rekursiv die Summe  $\sum_{i=1}^n i$  berechnen.

Summe( $n$ )  
1. if  $i = 0$  then return 1  
2. return  $n + \text{Summe}(n + 1)$

korrigiert:

Summe( $n$ )  
if  $n = 0$  then  
return 0  
else  
return Summe( $n - 1$ ) +  $n$

- b) Gegeben seien zwei natürliche Zahlen  $n$  und  $k$  mit  $n \geq k$ . Der folgende Algorithmus soll rekursiv den Binomialkoeffizienten  $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$  berechnen.

Binom( $n, k$ )

1. **return** Binom( $n - 1, k - 1$ ) + Binom( $n - 1, k$ )

Abbruch fehlt!

korrigiert:  
Binom( $n, k$ )  
if ( $k=n$ ) or ( $k=0$ )  
return 1

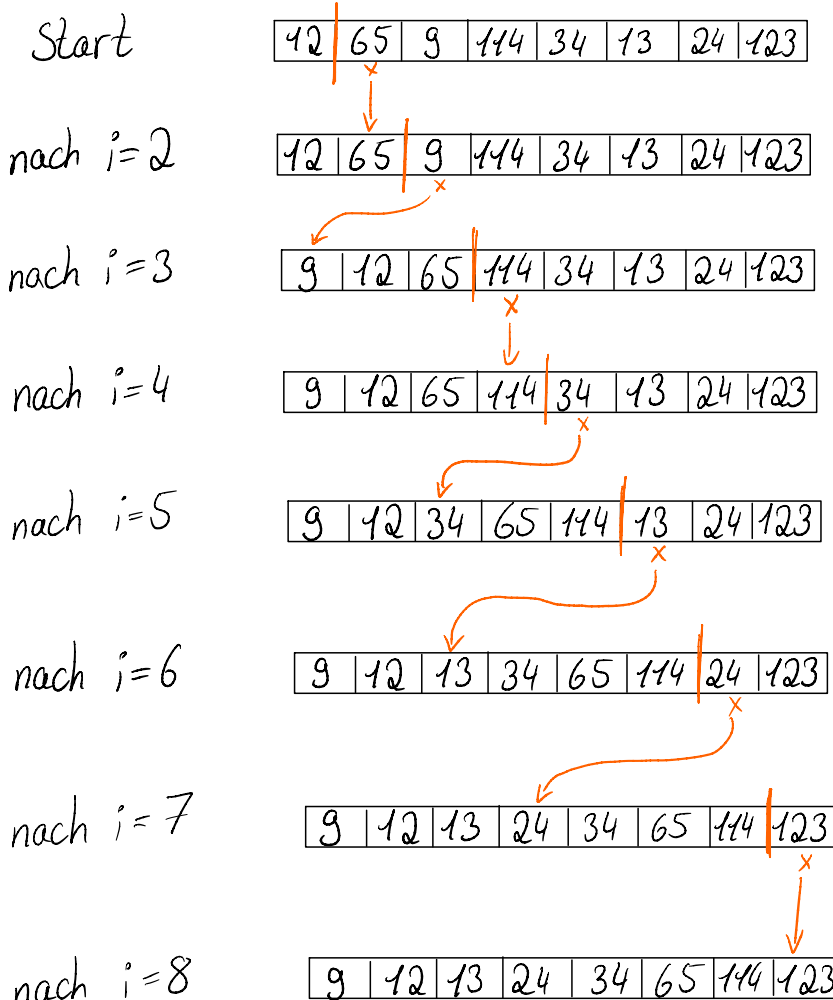
else  
return BinKoeff( $n-1, k-1$ )  
+ BinKoeff( $n-1, k$ )

#### Aufgabe 4 Insertion Sort (4 Punkte)

Gegeben sei das folgende Feld A:

12	65	9	114	34	13	24	123
----	----	---	-----	----	----	----	-----

Verwenden sie InsertionSort, um das Feld A zu sortieren. Geben Sie dabei den Zustand des Feldes nach jeder Iteration der for-Schleife an.



4.)

$\text{Prod}(n, a)$

if  $n = 0$

return 1

else

return  $a^n \cdot \text{Prod}(n-1, a)$

2.  
a)

$\text{MaxPair}(A, n)$

if  $n = 1$

return (1, 1)

else if  $n = 2$

return (1, 2)

else

$(i, j) = \text{MaxPair}(A, n-1)$

if  $|A[i] - A[j]| > |A[n-1] - A[n]|$

return (i, j)

else return (n-1, n)

wenn  $n = 1$  2 Zeitschritte

wenn  $n = 2$  3 Zeitschritte

wenn  $n > 2$  5 Zeitschritte + Rekursionsaufruf

b)

1 1 1

1

1 1

1

$1 \cdot T(n-1)$

2 in beiden Fällen

$n-2$  rekursive Aufrufe bis  $n=2$  gilt

$$\hookrightarrow (n-2) \cdot 5 + 3 = \underline{\underline{5n - 7}} \quad \text{wenn } n \geq 2$$