

17. Erstelle einen in Python geschriebenen Programmtext, um nach Eingabe einer natürlichen Zahl  $n$  die Folge  $g(n)$  für  $n = 0, 1, 2, \dots, n$  rekursiv zu berechnen und auszugeben:

Rekursionsanfang:  $g(0) = 0$   
Rekursionsvorschrift:  $g(n) = n - g(g(n-1))$ ,  $n > 0$

Teste das Programm für verschiedene Eingaben!

18. Der folgende in Python geschriebene Quelltext ist Teil des Algorithmus **SelectionSort**, der die  $n$  Komponenten  $a[0], a[1], \dots, a[n-1]$  einer Liste  $a$  aufsteigend sortiert:

```
j = 0
while j <= len(a)-2:
    min = a[j]
    for i in range(j+1, len(a)):
        if a[i] < min:
            min = a[i]
            a[i] = a[j]
            a[j] = min
    j += 1
```

Wir fassen die Anweisungen aus dem Schleifenrumpf der inneren Schleife dieses Programmauszugs gedanklich zum Anweisungsblock **A** zusammen  
(markiere Block **A** im obenstehenden Programmtext).

Um den Aufwand zu ermitteln, ein aus  $n$  Komponenten bestehendes array zu sortieren, fragen wir, wie oft Block **A** in Abhängigkeit von  $n$  abgearbeitet wird.

- a) Vervollständige die Einträge in folgender Tabelle, wobei  $z(j)$  angibt, wie oft Block **A** in Abhängigkeit von  $j$  abgearbeitet wird.

Index $j$	Index $i$	$z(j)$
$j = 0$	$\leq i \leq$	
$j = 1$	$\leq i \leq$	
$j = 2$	$\leq i \leq$	
....	....	....
$j = n-3$	$\leq i \leq$	
$j = n-2$	$\leq i \leq$	

- b) Die Gesamtzahl  $z$  der Abarbeitungen von Block **A** ergibt sich als

$$z = z(0) + z(1) + z(2) + z(3) + \dots + z(n-3) + z(n-2)$$

Vereinfache diese Summe und zeige so, daß  $z$  quadratisch mit  $n$  wächst!

Hinweis: Für die Summe der ersten  $n$  natürlichen Zahlen gilt bekanntlich:  
 $1 + 2 + \dots + n = \frac{1}{2} \cdot n \cdot (n + 1)$

19. Der auf die aus  $n$  Komponenten bestehende Liste  $a$  angewandte Sortieralgorithmus **MergeSort** ordnet die Inhalte dieser Komponenten so an, daß gilt:

$$a[0] \leq a[1] \leq \dots \leq a[n-1]$$

- a) Erläutere die Strategie von MergeSort in Worten; an welcher Stelle zeigt sich die rekursive Definition des Algorithmus?

- b) In der Programmiersprache Python wird mit dem Funktionsaufruf  
`sort(a, 0, n-1)`  
der Sortiervorgang veranlaßt.

Schreibe die Funktion `sort` als Python-Quelltext!

Anmerkung: Die Funktion `merge` ist als gegeben vorauszusetzen.

- c) Führe den Algorithmus MergeSort anhand einer 4-elementigen Liste schrittweise aus (Hinweis: statt `sort(a,0,3)` schreiben wir kürzer: `sort(0,3)`), ergänze fehlende Einträge:

sort(0,3)			
a[0]	a[1]	a[2]	a[3]
5	3	8	2

sort( , )		sort( , )	
a[0]	a[1]	a[2]	a[3]

sort( , )		sort( , )		sort( , )		sort( , )
a[0]		a[1]		a[2]		a[3]

a[0]		a[1]		a[2]		a[3]
merge(0,0,1)						merge( , , )

a[0]	a[1]	a[2]	a[3]
merge( , , )			

a[0]	a[1]	a[2]	a[3]

- d) Mit  $A(n)$  werde der Aufwand an elementaren Anweisungen bezeichnet, um eine aus  $n$  Komponenten bestehende Liste zu sortieren. Welche Funktionalgleichung lässt sich aus der Definition von MergeSort für  $A(n)$  ableiten (*beachte a) oder b)?*)?

Welche Funktion ist Lösung dieser Gleichung?