

# Übungsblatt 5 zur Vorlesung Algorithmen und Datenstrukturen (Sommer 2026)

**Abgabe:** Bis 2026-05-23 18:00, on ILIAS.

## 1. Aufgabe

30 Punkte

Wir betrachten drei hypothetische Algorithmen  $A_1$ ,  $A_2$ , und  $A_3$ , die jeweils auf Eingaben der Größe  $n$  arbeiten. Angenommen, Ihre Laufzeitmessungen für die Algorithmen ergeben die Ergebnisse in der Tabelle unten, bestimmen Sie für jeden Algorithmus, (1) ob die Laufzeit eine polynomiell beschränkte Funktion oder eine exponentiell wachsende Funktion ist, sowie (2) ein plausibles Modell für die Laufzeit als Funktion in  $n$ . Dokumentieren Sie jeweils, wie Sie zu Ihren Hypothesen gelangt sind.

$n$	$A_1$	$A_2$	$A_3$
1000	2.16	3.42	14.07
2000	10.12	10.80	20.09
3000	23.15	20.78	27.45
4000	39.45	31.68	40.87
5000	62.21	44.06	56.34
6000	89.55	59.20	79.93
7000	122.07	74.79	113.79
8000	160.33	91.47	159.65
9000	201.96	108.73	226.54
10000	249.62	127.12	319.91
11000	302.60	145.99	452.94
12000	359.69	167.10	639.38
13000	422.04	187.84	905.70
14000	489.27	208.60	1280.30
15000	562.19	232.19	1810.56
16000	639.55	256.01	2559.85
17000	721.89	279.81	3620.77
18000	810.27	304.60	5120.41
19000	901.62	331.36	7239.84

## 2. Aufgabe

20 + 20 + 20 Punkte

Für die folgenden drei Algorithmen, bestimmen Sie jeweils die  $\Theta$ -Klasse der Worst-Case-Laufzeit als Funktion in  $n$ . Nehmen Sie dabei als abstraktes Kostenmaß die Anzahl arithmetischer Operationen an.

$f(A[1..n])$ :

```
1  if  $n \leq 100$ 
2      return 100
3  end if
4   $ret := 0$ 
5  for  $i := 1, \dots, n$ 
6      for  $j := i + 1, \dots, n$ 
7           $ret := ret + A[i] \cdot A[j]$ 
8      end for
9  end for
10  $ret := ret + f(A[1.. \lfloor n/2 \rfloor])$ 
11  $ret := ret + f(A[\lfloor n/6 \rfloor .. \lfloor 2n/3 \rfloor])$ 
12  $ret := ret + f(A[\lfloor n/3 \rfloor .. \lfloor 5n/6 \rfloor])$ 
13  $ret := ret + f(A[\lfloor n/2 \rfloor .. n])$ 
14 return  $ret$ 
```

$g(n, x)$ :

```
1  if  $n \leq 100$ 
2      return 100
3  end if
4   $ret := 0$ 
5  for  $i := 1, \dots, \lfloor \sqrt{n} \rfloor$ 
6       $ret := ret + x \cdot i$ 
7  end for
8  return  $g(\frac{n}{4}, \frac{ret}{2}) + g(\frac{n}{4}, \frac{ret}{3}) + g(\frac{n}{4}, \frac{ret}{4})$ 
```

$h(A[1..n])$ :

```
1  if  $n \leq 100$ 
2      return 100
3  end if
4   $ret := 0$ 
5  for  $i := 1, \dots, \lceil \sqrt{n} \rceil$ 
6      for  $j := i^2, \dots, (i + 1)^2 - 1$ 
7           $ret := ret + A[i] \cdot A[j]$ 
8      end for
9  end for
10  $ret := ret + h(A[1.. \lfloor n/3 \rfloor])$ 
11 return  $ret$ 
```