

VL02, Aufgabe 1 (Übung)

Bestimmen Sie für die nachstehenden Prozeduren folgende Informationen:

- Berechnen Sie die Anzahl der Aufrufe von `tuwas()` für $n=100$
- Bestimmen Sie die Anzahl der Aufrufe von `tuwas()` als Funktion von n
- Bestimmen Sie die asymptotische Zeitkomplexität in $O(f(n))$ -Notation unter der Annahme, dass die asymptotische Zeitkomplexität von `tuwas()` $O(1)$ ist

<pre>static void proz1(int n) { for (int a=0; a<n; a++) tuwas(); for (int b=0; b<n; b++) tuwas(); }</pre>	<pre>static void proz2(int n) { for (int a=0; a<n; a++) for (int b=0; b<n; b++) tuwas(); }</pre>
<pre>static void proz3(int n) { for (int a=0; a*a<=n; a++) tuwas(); }</pre>	<pre>static void proz4(int n) { for (int a=1; a<=100; a++) for (int b=1; b<n*n; b++) for (int c=1; c<n; c++) tuwas(); }</pre>
<pre>static void proz5(int n) { for (int a=1; a<=n; a++) for (int b=1; b<=a; b++) tuwas(); }</pre>	<pre>static void proz6(int n) { while (n>=1) { tuwas(); n /= 2; } }</pre>

VL02, Aufgabe 2 (Übung)

Geben Sie bei jeder der untenstehenden Fragen an, ob die Aussage über die asymptotische Komplexität der angegebenen Funktion richtig ist oder nicht. Begründen Sie Ihre Antwort jeweils.

- 1) $7n \log_2 n = O(n)$
- 2) $n^3 + 2^n = O(2^n)$
- 3) $n^2 + 15n - 3 = O(n^3)$

VL02, Aufgabe 3 (Praktikum)

Diese Aufgabe dient dazu, Ihnen ein Gefühl für die Auswirkung der Zeitkomplexität eines Algorithmus auf die Laufzeit zu vermitteln.

- a) Messen Sie die Ausführungszeiten für die in der Klasse `Zeitmessung` angegebenen Methoden `func1`, `func2` und `func6` bei verschiedenen Werten für `n` (z.B. 100, 1000, 10000 und 100000). `func1` besitzt eine lineare, `func2` eine quadratische und `func6` eine logarithmische Laufzeit.

Hinweis: Benutzen Sie die Klasse `StopUhr`. Die Klasse finden Sie zusammen mit der Klasse `Zeitmessung` im ZIP-Archiv `UEB02.zip`.

```
import java.util.Date;

public class StopUhr
{
    private long startTime, stopTime;

    public void start()
    {
        startTime = System.nanoTime();
    }

    public void stop()
    {
        stopTime = System.nanoTime();
    }

    public long getDuration()
    {
        return stopTime - startTime;
    }
}
```

- b) Erweitern Sie die Klasse `Zeitmessung` um einen Zähler, so dass die Anzahl der Aufrufe der Methode `tuwas` für die Methoden `func1`, `func2` und `func6` bei verschiedenen Werten für `n` ebenfalls ausgegeben werden kann.