

# PT2: Serie 4

Abgabetermin	05.06.2018, 21:00 Uhr
Übungstermin	31.05.2018

---

## Aufgabe 1

Gegeben sind zwei Programme zur Matrizenmultiplikation, `multiply.c` und `multiply.java`. Bestimmen Sie die Komplexität der Multiplikation empirisch und theoretisch.

### Empirische Analyse

1. Ändern Sie die Programme so, dass Matrizen der Größen 10, 20, 30, 40, 50, 100, 500, 1000 Eingabe sind.
2. Verwenden Sie einerseits das Programm `/usr/bin/time` zur Ermittlung der Gesamtlaufzeit, und andererseits die C-Funktion `clock(3)` und die Javamethode `java.lang.System.nanoTime()` zur Ermittlung der Laufzeit von `mult`.
3. Protokollieren Sie Ihre Experimente. Geben Sie im Protokoll alle für die Messung relevanten Einflussparameter an.
4. Stellen Sie die Gesamtergebnisse tabellarisch oder graphisch dar.

### Theoretische Analyse

Zur theoretischen Analyse bestimmen Sie die Komplexitätsklasse der verwendeten Algorithmen, und erläutern Sie Ihre Analyse. Vergleichen Sie die theoretischen und die experimentellen Ergebnisse.

## Aufgabe 2

Bestimmen Sie, ob die folgenden Aussagen wahr sind. Beweisen Sie die wahren Aussagen.

- $O(n^2) \subseteq O(n^3)$
- $10 \cdot n + 20 \in O(n)$
- $n^{2/3} \in O(n \cdot \log n)$

## Abgabe

Reichen Sie Ihre Lösung in Form eines einzelnen gzip-komprimierten Tarfiles ein. Dieses soll im Wurzelverzeichnis Ihre Protokolle und Analysen als gesetzte PDF-Dateien enthalten.

Fügen Sie Anhänge zu den Protokollen, wie etwa modifizierten Quelltextdateien, in einen Unterordner `anhaenge` ein. Es sollen keine Binärdateien, Validationsskripte, Editorbackups, lokale Repositories, o.Ä. enthalten sein.

```
/
|--a1-empirisch.pdf
|--a1-theoretisch.pdf
|--a2.pdf
|--anhaenge
    |--[multiply*.c]
    |--[multiply*.java]
    |--[...]
```

## Zusatzaufgabe

Messen Sie für das C-Programm mithilfe von Hardware Performance Countern die Zahl der Takte (*cycles*), die zur Berechnung von `mult` benötigt werden. Auf einem nativ laufenden Linux können Sie dafür den Systemaufruf `perf_event_open(2)` (als root) verwenden.