

Vorlesung: PD Dr. Arne Meier  
Übung: Nicolas Fröhlich

Übungseinheit 13.10.23 - 18.10.23

# Datenstrukturen und Algorithmen

## 1. Übungsblatt (Python & O-Notation)

Relevant aus dem Skript: Vorlesung 1

### Eingangsfragen

Installieren Sie *Python3* und implementieren Sie ein Programm `hello_world.py`, welches den String "Hello, World!" in die Standardausgabe schreibt.

*Hinweis:* <https://www.python.org/about/gettingstarted/>

Gute Tutorials um Python zu lernen finden Sie beispielsweise hier:

<https://www.w3schools.com/python/>

<https://python.pages.doc.ic.ac.uk/cpp/lessons/cpp/>

<https://engineering.purdue.edu/~milind/datascience/2018spring/notes/lecture-2.pdf>

**Aufgabe 1**

Zeigen oder widerlegen Sie die folgenden Aussagen:

1.  $2n \in \mathcal{O}(n)$
2.  $n^2 \in \mathcal{O}(2^n)$
3.  $2^{2n} \in \mathcal{O}(2^n)$
4.  $\log_2(n^5) \in \mathcal{O}(n)$
5.  $2^n \in o(3^n)$
6.  $n^2 \in o(2n^2)$
7.  $2^n \in \Omega(n^{42})$
8.  $n \in \Omega(n^2)$

## Aufgabe 2

Analysieren Sie die Laufzeit der folgenden Algorithmen.

Geben Sie für jeden Schritt die Anzahl an Operationen in Abhängigkeit von  $n$  an und ordnen Sie die Algorithmen anschließend mithilfe der Landau Notation entsprechend ihrer asymptotischen Laufzeitkomplexität ein.

```
1 def algorithm_1(vector_1:list[int], vector_2:list[int]) -> int:
2     n = len(vector_1)
3     m = len(vector_2)
4     if n != m:
5         return None
6     result = 0
7     for i in range(n):
8         result += vector_1[i] * vector_2[i]
9     return result
```

```
1 def algorithm_2(input_list:list[int]) -> int:
2     n = len(input_list)
3     max = input_list[0]
4     for i in range(n):
5         list_sum = 0
6         for j in range(i, n):
7             list_sum += input_list[j]
8             if list_sum > max:
9                 max = list_sum
10    return max
```