

Übungsblatt 01

Aufgabe T1

Ordnen Sie die folgenden Funktionen in der Reihenfolge ihres asymptotischen Wachstums (ohne lange Nachzudenken). Mit $\log n$ bezeichnen wir den Logarithmus zur Basis 2.

- $\log(n)$
- n^2
- $\log(n^5)/\log \log(n)$
- $n \log n$
- $n^{\log \log n}$
- 2^n
- $\log(n)^{\log n}$
- \sqrt{n}

Aufgabe T2

Sei $x \in \mathbf{R}$ und $\lfloor x \rfloor = \max\{k \in \mathbf{Z} \mid k \leq x\}$.

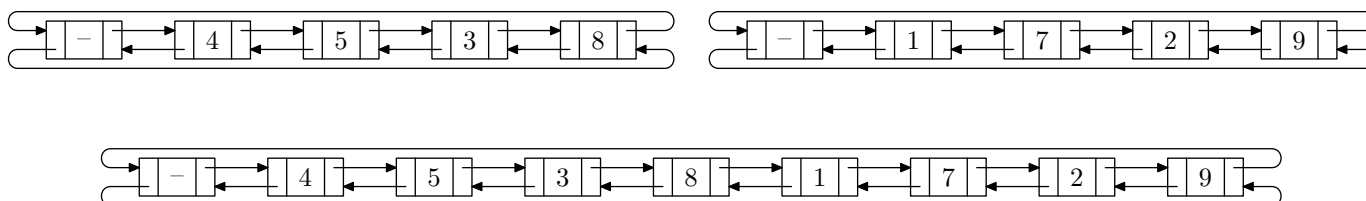
Zeigen Sie:

$$\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$$

Aufgabe T3

Entwerfen Sie einen einfachen Algorithmus, der zwei doppelt verkettete Listen aneinanderhängt und auf diese Weise eine neue doppelt verkettete Liste erzeugt. Können Sie dabei auf bedingte Verzweigungen verzichten?

Schreiben Sie den Algorithmus in Pseudocode oder in einer Programmiersprache nieder. Die folgende Grafik zeigt zwei Listen vor und nach der Verkettung.



Aufgabe T4

Beweisen oder widerlegen sie, dass $\frac{1}{2}n^2 + 7n + 14 = O(n^2)$ gilt.

Aufgabe H1 (10 Punkte)

Sei $x \in \mathbf{R}$ und $\lceil x \rceil = \min\{k \in \mathbf{Z} \mid k \geq x\}$.

Zeigen Sie:

$$\lceil x \rceil - 1 < x \leq \lceil x \rceil$$

Aufgabe H2 (10 Punkte)

Doppelt verkettete Listen wurden in der Vorlesung so implementiert:

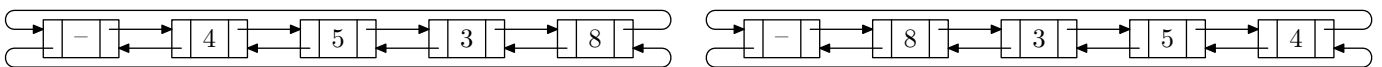
```
public class ADList<K, D> extends AbstractMap<K, D> {
    private Listnode<K, D> head;
    public ADList() {
        head = new Listnode<K, D>(null, null);
        head.pred = head;
        head.succ = head;
    }
    ...
}

public class Listnode<K, D> {
    K key;
    D data;
    Listnode<K, D> pred, succ;
    Listnode(K k, D d) { key = k; data = d; pred = null; succ = null; }
    ...
}
```

Implementieren Sie eine neue Methode *reverse()* in der Klasse *ADList*, welche die Reihenfolge in der Liste umkehrt. Dabei sollten Sie so vorgehen, dass kein zusätzlicher Speicher verwendet wird (abgesehen von konstant vielen Variablen), die Implementierung möglichst effizient ist und keine Rekursion verwendet wird.

Beachten Sie, dass Ihre Methode immer funktionieren muss, zum Beispiel auch mit einer leeren Liste.

Hier ist ein Beispiel, wie die Liste vor und nach dem Aufruf von *reverse()* aussehen wird:



Aufgabe H3 (10 Punkte)

Gegeben seien die Funktionen $f(n)$ und $g(n)$. Beweisen oder widerlegen Sie:

a) $O(f) \cdot O(g) = O(f \cdot g)$

b) Falls $g = O(f)$ und $h = O(f)$, dann gilt auch $g = O(h)$