

## Aufgaben Rekursion

### Aufgabe 1

Gegeben sei die folgende Funktion:

```
def was_tue_ich(zeichenkette):  
    if len(zeichenkette)==0:  
        return ""  
    else:  
        ab=1  
        anzahl=len(zeichenkette)-1  
        neue_zeichenkette=zeichenkette[ab:ab+anzahl]  
        return was_tue_ich(neue_zeichenkette)+zeichenkette[0]
```

1. Ermittle, welches Wort bei folgenden Aufrufen erzeugt wird:  
    `was_tue_ich('ei');`  
    `was_tue_ich('tier');`
2. Weise im Quelltext die drei Eigenschaften der Rekursion nach.

### Aufgabe 2 – Aus der Mathematik

Eine Funktion sei definiert durch:

$$f(n) = \begin{cases} f(n-1) + 3n(n-1) + 1 & \text{für } n > 1 \\ 1 & \text{sonst} \end{cases}$$

- a) Fülle die Wertetabelle aus:

n	1	2	3	4	5	6
f(n)						

- b) Beschreibe den Zusammenhang zwischen n und f(n).
- c) Schreibe eine rekursive Funktion, die den Funktionswert  $f(n)$  berechnet.
- d) Schreibe eine Funktion, die Funktionswert  $f(n)$  berechnet, ohne Rekursion zu verwenden.  
Nutze eine Schleife!

### Aufgabe 3 – „Lineal“ Algorithmus

Ziel:

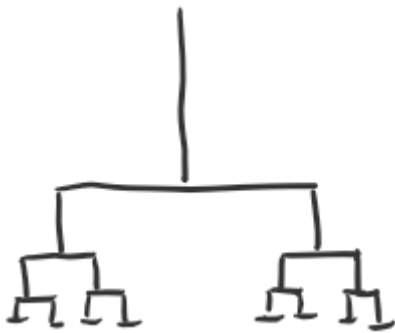


Zeichnen eines „Lineals“. Überlege dir zunächst eine rekursive Strategie und implementiere.

Eingaben: Anfang, Ende, Starthöhe

### Aufgabe 4 – Grafik 1

Entwerft ein rekursives Programm zum Zeichnen eines binären Baumes:



### Aufgabe 6 - Zusatz

Gegeben sei die folgende Funktion, die sogenannte Ackermann-Funktion:

$$a(0, m) = m + 1$$

$$a(n + 1, 0) = a(n, 1)$$

$$a(n + 1, m + 1) = a(n, a(n + 1, m))$$

Berechne „von Hand“ den Funktionswert von  $a(1, 2)$ .

Implementiere die Ackermann-Funktion in Python.