Hochschule
Albstadt-Sigmaringen
Albstadt-Sigmaringen University

Technische Informatik
Prof. Dr. Tobias Häberlein & Sebastian Rädle

Algorithmik
SS 12

Aufgabenblatt 1

(Bearbeitungszeit: 3 Praktikumstermine)

Aufgabe 1.0. **10 Punkte**

Schreiben Sie eine rekursive Funktion ...

(a) ... *sumRek*(n), die die Summe der ersten n natürlichen Zahlen berechnet.

(b) ... *lenRek*(l), die die Länge der Liste l berechnet.

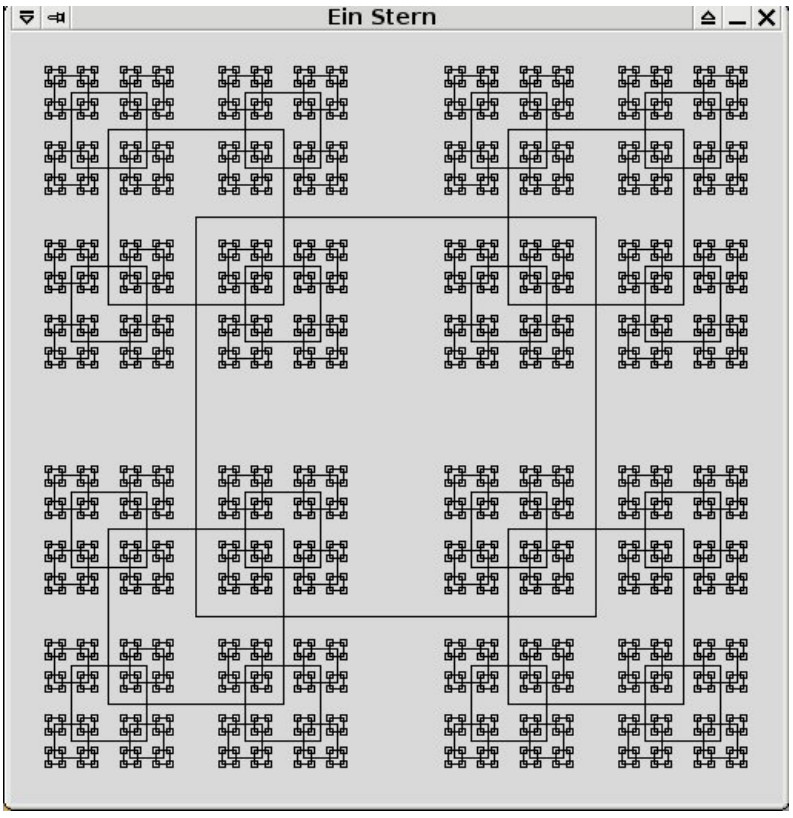
(c) ... *anzBlanks*(s), die die Anzahl der Leerzeichen in einem String s berechnet.

(d) ... *inserts*(x, l), die alle möglichen Einfügungen des Elements x in die Liste l zurückliefert. Beispiel:

```
>>> inserts('a',[1,2,3])
[['a',1,2,3],[1,'a',2,3],  [1,2,'a',3],  [1,2,3,'a']]
```

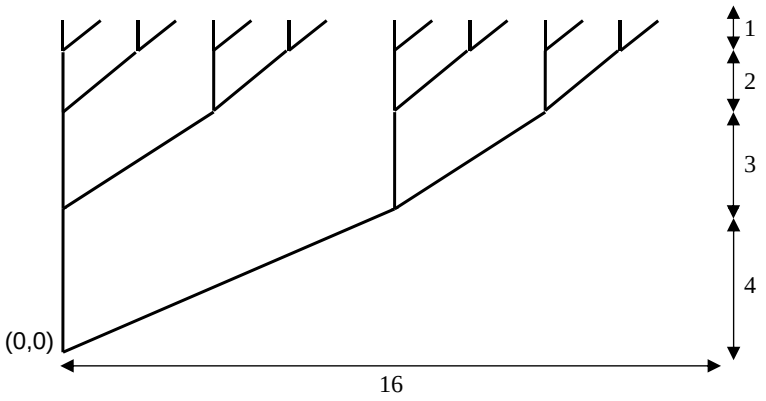
Aufgabe 1.1. **10 Punkte**

Zeichnen Sie durch eine rekursiv definierte Python-Funktion und unter Verwendung der *graphics*-Bibliothek folgenden Stern:



Aufgabe 1.2. **10 Punkte**

Schreiben Sie eine rekursive Prozedur *baum*(*x*,*y*,*b*,*h*) zum Zeichnen eines (binären) Baumes derart, dass die Wurzel sich bei (x, y) befindet, der Baum b breit und h hoch ist. Definieren Sie hierzu eine Python-Prozedur *line*(*x1*,*y2*,*x2*,*y2*), die eine Linie vom Punkt (*x1*,*y2*) zum Punkt (*x2*,*y2*) zeichnet. Folgende Abbildung zeigt ein Beispiel für die Ausgabe die der Aufruf *baum*(0,0,16,4) erzeugt.



Aufgabe 1.3. **15 Punkte**

(a) Kombinieren Sie Insertion Sort und Quicksort folgendermaßen: Listen der Länge größer 100 werden mittels Quicksort sortiert; der Rekursionsabbrucht erfolgt (nicht bei Länge 1) sondern bei einer Länge kleiner als 100; diese Listen sollen dann über Insertion Sort „fertig“ sortiert werden.

(b) Verwenden Sie das *timeit*-Modul, um Insertion-Sort, Quicksort und die in Teilaufgabe (a) programmierte Kombination der beiden für zufällige Listen der Länge 100,1000 und 1 Mio zu testen.

(c) Verwenden Sie das *timeit*-Modul, um Insertion-Sort, Quicksort und die in Teilaufgabe (a) programmierte Kombination der beiden für zufällige Listen der Länge 100,1000 und 1 Mio zu testen.