

Praktikum Aufgabe 8 (18 Punkte)

Informatik für den Studiengang Physik
Dr. Herbert Süße, Institut für Informatik
14.6.2007 Abgabe: Bis 2. Juli 2007

Nutzung rekursiver Funktionen

1 Aufgabe 8a (14 Punkte)

Es sind zwei Versionen einer **rekursiven** Funktion zur Multiplikation zweier Matrizen zu schreiben. In der Vorlesung ist die Rekursionsidee dargestellt worden. Es sollen ausschließlich quadratische Matrizen sein, deren Dimensionen eine Potenz von 2 sind.

Schreiben Sie für die rekursive Funktion zwei Versionen:

- a) Übergeben werden die Matrizen und weitere benötigte Parameter. In der Funktion sind die Teilmatrizen (der zu multiplizierenden Matrizen A und B) mit dynamischer Speicherplatzzuweisung neu anzulegen.
- b) Überlegen Sie, wie Sie den Funktionsheader gestalten müssen, d.h. welche Parameter im Gegensatz zu a) benötigt werden, damit in der Funktion nicht neue Matrizen angelegt werden müssen, sondern direkt der Speicher der übergebenen Matrizen A und B genutzt werden kann (in situ Algorithmus).

Schreiben Sie weiterhin eine Funktion, die zwei Matrizen wie "üblich" multipliziert. Im Hauptprogramm testen Sie alle drei Funktionen und vergleichen die Ergebnisse zur Kontrolle.

2 Aufgabe 8b (4 Punkte)

Gegeben sei folgende rekursive Funktion in Pseudocode:

```
procedure fraktal(integer x,y,r);  
begin  
if  $r > 0$  then  
    begin  
        fraktal(x-r,y+r,r div 2);  
        fraktal(x+r,y+r,r div 2);  
        fraktal(x-r,y-r,r div 2);  
        fraktal(x+r,y-r,r div 2);  
        box(x,y,r);  
    end  
end
```

Diese rekursive Funktion enthält als einzige Hilfsfunktion die Funktion `box`, die ein Quadrat mit dem Mittelpunkt x, y und der Seitenlänge von $2r$ zeichnet, wobei der Rand schwarz ist und das "Innere" weiß ist (Sie müssen das Innere explizit weiß setzen und den Rand schwarz). Setzen Sie diese rekursive Funktion in C/C++ um, erläutern Sie was in der Funktion passiert, rufen Sie diese Funktion auf mit geeigneten Parametern und lassen Sie sich überraschen, was Ihre Funktion grafisch bewirkt.