

Programmierung 2 - Sommersemester 2020

Prof. Dr. Peter Birkner

Übungsblatt Nr. 15 Abgabe KW 19

1. Aufgabe

Mithilfe des Algorithmus von Euklid kann der größte gemeinsame Teiler (ggT) zweier Zahlen sehr einfach rekursiv berechnet werden. Nach Euklid gilt für zwei natürliche Zahlen a und b folgendes:

$$\begin{aligned}\text{ggT}(a, 0) &= a \\ \text{ggT}(a, b) &= \text{ggT}(b, a \bmod b)\end{aligned}$$

Implementieren Sie ein Programm, welches den ggT für zwei als Argumente übergebene natürliche Zahlen rekursiv berechnet. Erweitern Sie dazu die Klasse MathFunctions aus den Übungen 3 und 4 um folgende Methode: `long berechneGgt(long a, long b)`.

2. Aufgabe

Palindrome sind Worte oder allgemein Zeichenketten, die von links als auch von rechts gelesen identisch sind. Beispiel sind Otto, Reittier oder Rentner.

- (a) Implementieren Sie ein Programm, welches für einen gegebenen Eingabestring feststellt, ob dieser ein Palindrom ist oder nicht. Zur Eingabe des zu untersuchenden Strings soll ihr Programm zwei Möglichkeiten bieten. Zum einen soll der Eingabestring beim Programmaufruf als Argument übergeben werden können. Zum anderen soll es aber auch möglich sein, den Eingabestring aus einer Textdatei zu lesen.

Das Problem kann sowohl iterativ als auch rekursiv gelöst werden. Implementieren Sie beide Lösungen. Entwerfen Sie ihr Programm so, dass es leicht um weitere Algorithmen zur Lösung des Problems erweiterbar ist.

Beachten Sie unbedingt folgende Implementierungsdetails:

- Erzeugen Sie ein Interface Palindrom mit der Methode `boolean istPalindrom(String wort)`.
 - Nennen Sie die erste Implementierung PalindromRekursiv.
 - Nennen Sie die zweite Implementierung PalindromIterativ.
- (b) Implementieren Sie eine kleine Messreihe, welche die Geschwindigkeit der Berechnung für verschiedene Problemgrößen misst und als Comma-separated values in eine Textdatei schreibt.
- (c) Führen Sie Messungen für ihre iterative und ihre rekursive Lösung durch und tragen beide Messreihen in einem Diagramm gegeneinander ab. Zum Beispiel so wie in Abbildung 1 dargestellt. Entscheiden Sie selbst, bis zu welcher Problemgröße Sie Messungen durchführen. Bei der Präsentation ihrer Lösung sollten Sie in der Lage sein, die gemessenen Effekte zu erklären.

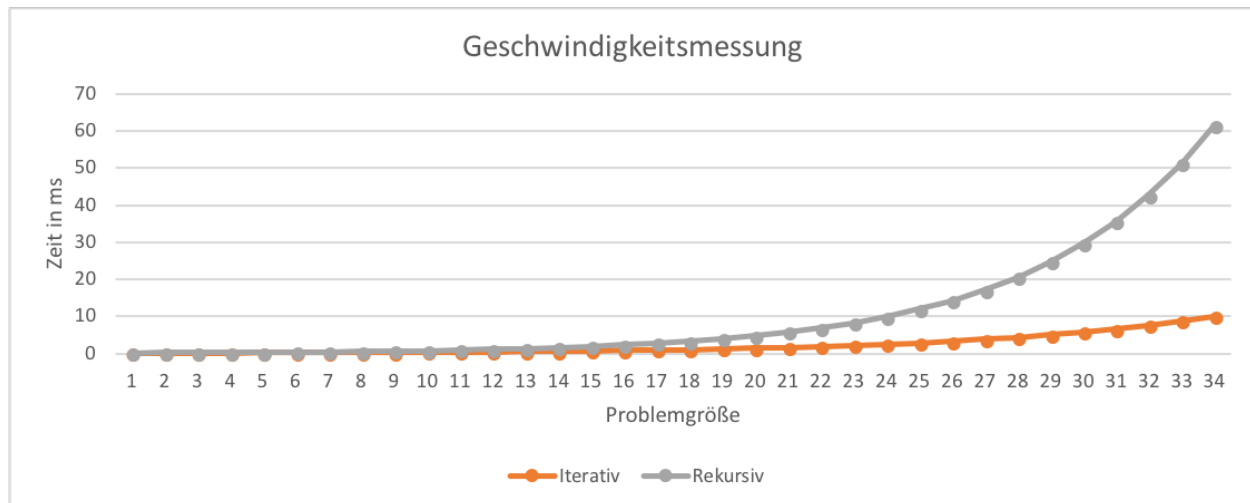


Abbildung 1: Beispiel für eine Messreihe zum Vergleich der iterativen und der rekursiven Implementierung.

Hinweise:

- Bei der Geschwindigkeitsmessung kann die Klasse `System.nanoTime()` hilfreich sein.
- Zur Erzeugung des Diagramms in Teilaufgabe c kann ein beliebiges Tabellenkalkulationsprogramm verwendet werden. Eine andere Möglichkeit, die jedoch mehr Einarbeitungsaufwand erfordert, ist beispielsweise die Nutzung von Gnuplot.