

Messreihe zum Vergleich der rekursiven und iterativen Palindrom-Lösung

A. Venet und H. Schall

02.05.2020

1 Code

Um die Messreihe durchzuführen wurde die Klasse `Messreihe` erstellt. In dieser wurden zwei `String[] []` Arrays erstellt. Das Array `palindromicStrings` enthält 34 Arrays, die je 4 Palindrome enthalten (das erste der 34 mit Problemgröße 1, das zweite mit Größe 2, usw.). Das Array `nonPalindromicStrings` ist genau gleich aufgebaut, bis auf den Umstand, dass es ausschließlich nicht-palindromische Strings enthält.

```
// Auszug aus Messreihe.java
private static void process(String[] [] testList, String
    rekursivNameFile, String iterativNameFile) {
    StringBuilder it = new StringBuilder();
    StringBuilder rk = new StringBuilder();
    long beginn;
    long ende;
    long rekursivZeit;
    long iterativZeit;
    for (int j = 0; j < testList.length; j++) {
        rekursivZeit = 0L;
        iterativZeit = 0L;
        for (int k = 0; k < testList[j].length; k++) {
            beginn = System.nanoTime();
            palindromRekursiv.istPalindrom(testList[j][k]);
            rekursivZeit += System.nanoTime() - beginn;
            beginn = System.nanoTime();
            palindromIterativ.istPalindrom(testList[j][k]);
            iterativZeit += System.nanoTime() - beginn;
        }
        it.append(iterativZeit/4 + ", ");
    }
}
```

```

        rk.append(rekursivZeit/4 + ", ");
    }
    String itString = it.toString();
    itString = itString.substring(0, itString.length() - 2);
    String rkString = rk.toString();
    rkString = rkString.substring(0, rkString.length() - 2);

    try {
        FileWriter rekursivWriter = new
            FileWriter(rekursivNameFile);
        rekursivWriter.write(rkString);
        FileWriter iterativWriter = new
            FileWriter(iterativNameFile);
        iterativWriter.write(itString);
        rekursivWriter.close();
        iterativWriter.close();
        System.out.println("Erfolgreich gemessen, siehe " +
            rekursivNameFile + " und " + iterativNameFile);
    } catch (IOException e) {
        System.out.println("Fehler!");
        e.printStackTrace();
    }
}

```

2 Ergebnisse

Die Berechnungsdauern der Strings aus dem Array `nonPalindromicStrings` variieren sehr stark. Ein Zusammenhang zwischen Dauer und Problemgröße wurde nicht ersichtlich. Deshalb umfasst die folgende Auswertung lediglich die Berechnung des Ergebnisses von Palindromen (Ergebnis `true`).

2.1 tabellarisch

Diese Tabelle zeigt die durchschnittliche Dauer der Berechnung in Nanosekunden. Die Problemgröße stellt die Größe des Eingabestrings dar. Für jede Größe wurde ein Durchschnitt aus der Dauer der Berechnung von vier Palindromen errechnet.

Problemgröße	1	2	3	4	5	6	7	8	9	10	11
iterativ	1125	1025	850	1075	1125	1350	1275	1550	1425	1700	1725
rekursiv	2025	4350	1050	1475	2375	3500	1900	2550	2275	3725	3875

12	13	14	15	16	17	18	19	20	21	22	23	24
1875	1825	2150	2175	2500	2200	2500	1950	1350	2075	1525	1200	4575
3425	3225	4725	6525	5475	4425	4925	4250	4325	6050	6425	7050	10775

25	26	27	28	29	30	31	32	33	34
8350	3225	3325	3500	3550	3700	10700	3375	4325	6125
11675	6375	7875	8050	8250	8475	10900	8625	9600	10975

2.2 als Diagramm

