Algorithmen und Datenstrukturen
Übungszettel 13

Martin Avanzini <martin.avanzini@uibk.ac.at>
Thomas Bauereiß <thomas.bauereiss@uibk.ac.at>
Herbert Jordan <herbert@dps.uibk.ac.at>
René Thiemann <rene.thiemann@uibk.ac.at>

21. Juni, zur Besprechung am 28. Juni

Aufgabe 1) Knuth-Morris-Pratt

1. Verwenden Sie den Algorithmus von Knuth, Morris und Pratt zum Suchen der nachfolgenden Muster w in Texten. Geben Sie das Array `next` in der Form des endlichen Automaten $\mathcal{B}'_w$ an. Zeigen Sie einen akzeptierenden Lauf für den gegebenen Suchtext s.

   (a) $w =$ ABCD und $s =$ ABCABCD.

   (b) $w =$ ACAB und $s =$ ACACAB.

   (c) $w =$ 0110 1110 1101 und $s =$ 0110 1101 1100 1101 1101 1011 0111 0110 0111 1011 0100.

2. Implementieren Sie den Algorithmus von Knuth, Morris und Pratt in Java. Betrachten Sie hierfür folgendes Template:

```java
public class Kmp {

    /**
     * @param das zu suchende Wort
     */
    public Kmp(String w) {
      throw new RuntimeException("zu implementieren!");
    }

    /**
     * @param  der Suchtext
     * gibt am Bildschirm die Position des ersten Vorkommens
     * des zu suchenden Wortes w im Suchtext s aus
     */

    void find(String s) {
    throw new RuntimeException("zu implementieren!");
    }

    /**
```

```
21        * @param  der Suchtext
22        * gibt am Bildschirm die Position aller Vorkommen
23        * des zu suchenden Wortes w im Suchtext s aus
24        */
25      void findAll(String s) {
26        throw new RuntimeException("zu implementieren!");
27      }
28
29      public static void main(String[] args) {
30      Kmp k = new Kmp(args[0]);
31      k.find(args[1]);
32      System.out.println("");
33      k.findAll(args[1]);
34      }
35  }
```

Der Konstruktor `Kmp(String w)` führt die Kompilierung des zu suchenden Wortes `w` durch.

(a) Implementieren Sie die Methode `void find(String s)` welche wie in den Vorlesungsunterlagen das erste Vorkommen des gesuchten Wortes `w` am Bildschirm ausgibt.

(b) Modifizieren Sie (in der Methode `void findAll(String s)`) obige Lösung so dass *alle* Vorkommen des gesuchten Wortes `w` am Bildschirm in Zeit $\mathcal{O}(n + m)$ ausgegeben werden. Hier bezeichnet n und m die Länge des Suchtextes `s` bzw. des Wortes `w`.

Aufgabe 2) Boyer-Moore Wenden Sie Boyer-Moore auf die Beispiele 1a und 1b an.

Aufgabe 3) Klausurvorbereitung Bereiten Sie sich gründlich auf die Klausur vor. Stellen Sie etwaige Fragen an diesem letzten Übungstermin.