

## 1. Übungsblatt

**Ausgabe:** 25.04.2008    **Abgabe:** 09.05.2008, vor der Vorlesung

### Aufgabe 1: Exakte Suche in Texten

10 Punkte

Entwerfen Sie eine Variante des einfachen Algorithmus zur exakten Suche in Texten (mit Fenstervergleichen von links nach rechts), der alle Vorkommen eines Wortes  $s$  in einem Wort  $t$  findet. Implementieren Sie dazu in der Klasse `SearchClass` die virtuelle Methode `searchAllStringsFromLeftToRight`:

```
public class SearchClass {  
    public static int[] searchAllStringsFromLeftToRight(String s, String t){  
        :  
    }  
}
```

Fügen Sie bitte auch erläuternde Kommentare ein.

### Aufgabe 2: Algorithmus von Boyer und Moore

10 Punkte

Wie viele Vergleiche von Buchstaben benötigt der Algorithmus von Boyer und Moore (mit *bad character rule*), um das Wort AATACG im Wort AACACTACGCGAATACGC zu finden? Verwenden Sie dazu die folgende Verschiebungstabelle:

	0	1	2	3	4	5	6
A	1	1	1	2	1	2	3
C	1	2	3	4	5	1	2
G	1	2	3	4	5	6	1
T	1	2	3	1	2	3	4

Dabei finden Sie zum Beispiel den Eintrag  $S(3, G)$  in der Zeile  $G$  und der Spalte 3, d.h.  $S(3, G) = 4$ .

Veranschaulichen Sie auch den Ablauf des Algorithmus.

### Aufgabe 3: Algorithmus von Knuth, Morris und Pratt

10 Punkte

Bestimmen Sie für das Wort 10100101001001 die Tabelle mit den Längen der eigentlichen Ränder aller Präfixe wie im Algorithmus von Knuth, Morris und Pratt beschrieben.