

## Übungsblatt 2

Abgabe bis zum 19. Mai 2009

### Aufgabe 6

mündlich

Ein String  $x = x_1 \cdots x_n$ ,  $n \geq 1$ , heißt  $k$ -periodisch, falls  $x_i = x_{i+k}$  für  $i = 1, \dots, n - k$  gilt. Die Zahl  $P(x) = \min \{k \geq 1 \mid x \text{ ist } k\text{-periodisch}\}$  heißt die Periode von  $x$ .

- Bestimmen Sie die Periode für die Wörter  $x = \text{laola}$  und  $y = \text{olalaolala}$ .
- Geben Sie einen Linearzeit-Algorithmus an, der  $P(x)$  berechnet. Begründen Sie.
- Zeigen Sie, dass  $P(x)$  genau dann ein von  $|x| = n$  verschiedener Teiler von  $n$  ist, wenn es ein Wort  $y$  und eine Zahl  $l > 1$  gibt mit  $x = y^l$ .

### Aufgabe 7

mündlich

**Bubble-Sort** sortiert eine Zahlenfolge  $a_1, \dots, a_n$  durch wiederholtes Vertauschen von benachbarten Folgengliedern:

#### Bubble-Sort

```
1 Input: Eine Zahlenfolge  $a_1, \dots, a_n$ 
2 for  $i := n - 1$  downto 1 do
3   for  $j := 1$  to  $i$  do
4     if  $a_j > a_{j+1}$  then vertausche( $a_j, a_{j+1}$ )
5 Output:  $a_1, \dots, a_n$ 
```

- Wie verarbeitet **Bubble-Sort** die Eingabefolge 3, 6, 1, 7, 9, 2, 4, 8?
- Finden und beweisen Sie eine geeignete Invariante für die innere **for**-Schleife.
- Benutzen Sie die Schleifeninvariante aus (b) für den Nachweis einer geeigneten Invariante für die äußere **for**-Schleife.
- Beweisen Sie die Korrektheit von **Bubble-Sort** mithilfe der Invariante aus (c).
- Bestimmen Sie die Anzahl an Vergleichen, die **Bubble-Sort** im besten und im schlechtesten Fall benötigt.
- Verbessern Sie die bestcase-Komplexität von **Bubble-Sort**, indem Sie die äußere **for**-Schleife durch eine **repeat**-Schleife ersetzen, die für möglichst wenige Werte von  $i$  durchlaufen wird und insbesondere abbricht, sobald die Folge sortiert ist.

### Aufgabe 8

mündlich

Implementieren Sie **MergeSort** als nichtrekursives »in place«-Sortierverfahren.

*Hinweis:* Die zu sortierende Zahlenfolge soll durch einen Zeiger auf eine einfach verkettete Liste  $L$  übergeben werden. Bei jedem Schleifendurchlauf soll der Algorithmus die Liste  $L$  zuerst so in zwei Listen  $A$  und  $B$  zerlegen, dass sortierte Teilstücke maximaler Länge (so genannte *Runs*) zusammenbleiben und abwechselnd auf die beiden Listen  $A$  und  $B$  verteilt werden. Im zweiten Teil der Schleife sollen die Listen  $A$  und  $B$  wieder zu einer Liste gemischt werden, so dass sich die Anzahl der *Runs* bei jedem Schleifendurchlauf halbiert.

### Aufgabe 9

mündlich

Die Prozedur **Merge** benötigt im schlechtesten Fall  $n + m - 1$  Vergleiche, um zwei sortierte Zahlenfolgen  $a_1 \leq \dots \leq a_n$  und  $b_1 \leq \dots \leq b_m$  zu einer sortierten Folge  $c_1 \leq \dots \leq c_{n+m}$  zusammenzufügen.

- Zeigen Sie, dass es keine Prozedur gibt, die dies im Fall  $n = m$  mit weniger als  $2n - 1$  Vergleichen schafft.
- Überlegen Sie sich ein Verfahren, das im Fall  $m \ll n$  (z.B.  $m \approx \sqrt{n}$ ) mit deutlich weniger Vergleichen auskommt.

### Aufgabe 10

mündlich

Konstruieren Sie einen vergleichsbasierten Algorithmus, der im schlechtesten Fall eine möglichst geringe Anzahl  $V(n)$  an Vergleichen benötigt, um

- das Maximum und das Minimum,
- das größte und zweitgrößte Element

einer Folge von  $n$  Zahlen zu finden.

*Hinweis:* Für (a) sind  $\lceil 3n/2 \rceil - 2$  und für (b) sind  $n + \lceil \log_2 n \rceil - 2$  Vergleiche optimal.

### Aufgabe 11

6 Punkte

Bestimmen Sie die minimale Anzahl an Vergleichen, die ein vergleichsbasierter Algorithmus im besten Fall benötigt, um

- eine Folge von  $n$  Zahlen zu sortieren, (mündlich)
- den Median einer Folge von  $n$  Zahlen zu finden, (2 Punkte)
- das Maximum einer Folge von  $n$  Zahlen zu finden, (2 Punkte)
- das Maximum und das Minimum einer Folge von  $n$  Zahlen zu finden. (2 Punkte)

### Aufgabe 12

4 Punkte

Ein String heißt *zyklische Überdeckung* von  $y$ , falls  $y$  ein Teilwort eines Wortes in  $\{x\}^*$  ist. Geben Sie einen Linearzeit-Algorithmus an, der für zwei Texte  $x$  und  $y$  entscheidet, ob  $x$  eine zyklische Überdeckung von  $y$  ist. Begründen Sie.