

6. Präsenzübung „Algorithmen und Datenstrukturen“

Sommersemester 2009

1 Korrektheit

a) Formulieren Sie für die While-Schleife folgende Variante von Bubblesort eine Invariante.

Die Invariante kann (möglichst präzise) umgangssprachlich formuliert sein, soll aber dafür geeignet sein, die Korrektheit des Sortieralgorithmus nachzuweisen.

```
static void bubblesortUpDown(int[] a) {
    int n = a.length;
    i = 0;
    while (i < n / 2) {
        for (int j = i; j <= n - i - 2; ++j) {
            swapIfNeeded(a, j, j + 1);
        }
        for (int j = n - i - 3; j >= i; --j) {
            swapIfNeeded(a, j, j + 1);
        }
        ++i;
    }
}

static void swapIfNeeded(int[] a, int i, int j) {
    if (a[i] > a[j]) {
        int h = a[i];
        a[i] = a[j];
        a[j] = h;
    }
}
```

b) Betrachten Sie folgenden Programmabschnitt

```
k = 0;
i = 2;
while (n > 1) {
    if (n % i == 0) {
        a[k] = i;
        k = k + 1;
    }
}
```

```

        n = n / i;
    } else {
        i = i + 1;
    }
}

```

der eine natürliche Zahl n in Primfaktoren zerlegt und stellen Sie eine Invariante für einen Beweis der partielle Korrektheit bezüglich der folgenden Spezifikation auf:

- Vorbedingung: $\{n = A \geq 1\}$
- Nachbedingung: $\{a[0] \cdot a[1] \cdot \dots \cdot a[k - 1] = A\}$

2 Abstrakte Datentypen

1. Was ist ein abstrakter Datentyp? Worin unterscheiden sich abstrakte Datentypen und Datenstrukturen?
2. In Java gibt es die Möglichkeit, *abstrakte Klassen* zu definieren. Beschreibt jede solche abstrakte Klasse einen abstrakten Datentypen? Begründen Sie.

3 Untere Schranke

Zeigen Sie, dass der Algorithmus der binären Suche bezüglich seiner Laufzeit optimal ist. Genauer gesagt: Beweisen Sie, dass jeder vergleichsbasierte Algorithmus für die Suche nach einem Element x in einem sortierten Array der Länge n die Laufzeit $T(n) = \Omega(\log n)$ hat. Beachten Sie, dass der Algorithmus $n + 1$ verschiedene Ergebnisse liefern kann:

- x befindet sich auf Position 0
- x befindet sich auf Position 1
- ...
- x befindet sich auf Position $n - 1$
- x ist nicht im Array enthalten