

Übung zu Algorithmen und Programmieren III, WS 2001/2

Übung 4

Ausgabe: 6.11.01

Abgabe: 15.11.01 bis 14.00 Uhr

Aufgabe 1 (6 P)

Für ein Feld $b[0..n-1]$ bestehend aus n Zahlen soll die Summe der Zahlen berechnet werden. Formulieren Sie die Voraussetzung und den Effekt für dieses Problem. Geben Sie die Invariante und die Beschränkungsfunktion für die `while`-Schleife des folgenden Programmstücks an:

```
s = 0;
i = 0;
while (i < n) {
    s = s + b[i];
    i = i + 1;
}
```

Aufgabe 2 (6 P)

Entwickeln Sie bitte schrittweise eine Anweisungsfolge S zum Lösen des folgenden Problems: Gegeben seien n Objekte ($B[0..1]$, $n \geq 0$ fix), jedes Objekt hat eine der drei Farben: rot, weiß, blau. Sortiere die Objekte so, dass die Folge mit diesen Objekten aus drei Regionen besteht. In jeder Region kommen Objekte nur mit einer Farbe vor. Hierbei ist die Reihenfolge der Regionen die folgende: rot, weiß, blau. Für S sollen die Voraussetzung

$$\{n \geq 0 \wedge b = B\}$$

und der Effekt

$$\{\exists 0 \leq Lo, Hi < n: b[0..Lo-1] \text{ ist rot} \wedge b[Lo..Hi-1] \text{ ist weiß} \wedge b[Hi..n] \text{ ist blau}\}$$

gelten.

Aufgabe 3 (3 P)

Können Sie die Terminierung des folgenden Algorithmus beweisen?

```
{n ≥ 1}
while (n != 1)
    if odd(n) then 3*n+1 else n = n/2;
{true}
```

Wenn Sie keine Terminierungsfunktion finden, schauen Sie sich die Ausführung für einige n , z.B. $n = 15$ oder $n = 27$, an. Das gibt zumindest einen Eindruck, warum das Problem nicht trivial ist. Diskutieren Sie das Problem und geben Sie die mathematische Formulierung für das Problem an.

Aufgabe 4 (10 P)

a) Entwickeln Sie eine Anweisungsfolge S zum "Umdrehen" eines Feldes a mit $a.length = n$, $n \geq 1$. Die Feldelemente sind ganze Zahlen. S soll der Spezifikation genügen:

$$\{n \geq 1 \wedge a = A\} \ S \ \{ \forall 0 \leq i < n : a[i] = A[n-i-1] \}$$

S soll durch eine Schleife realisiert werden, die das erste mit dem letzten Element vertauscht, dann das zweite mit dem vorletzten usw. Verwenden Sie kein Hilfsfeld.

b) Kann man auch ohne Hilfsvariable auskommen?

c) Geben Sie eine geeignete Invariante und Schleifenbedingung an und zeigen Sie die partielle Korrektheit.

Bitte wenden

d) Zeigen Sie die totale Korrektheit von S.

Aufgabe 5 (4 P)

Erweitern Sie bitte die Datenstruktur aus der Aufgabe 4 des 3. Übungsblattes um die folgenden Funktionen:

```
public int showMaxPrio (); // gibt die höchste Priorität zurück
public void compress();   // Entfernt alle Duplikate
                          // (element1 == element2) aus der PrioQueue
```

Geben Sie für die Funktionen jeweils eine informelle Spezifikation an.