

7. Übungsblatt

1. Aufgabe (4 Punkte)

- a) Verändern Sie die **majority**-Funktion aus der Musterlösung (siehe unsere Veranstaltungsseite), indem Sie zuerst die **for**-Schleifen in eine **while**-Schleifen verwandeln.
- b) Postulieren Sie möglichst aussagekräftige Invarianten für beide Schleifen und testen Sie diese mit Hilfe der **assert**-Anweisungen von Python.

2. Aufgabe (6 Punkte)

- a) Postulieren Sie eine möglichst aussagekräftige Invariante für die Schleife der **partition**-Funktion des Quicksort-Algorithmus (erste Lösungsversion) aus unseren Musterlösungen und testen Sie diese mit Hilfe der **assert**-Anweisungen von Python.
- b) Schlagen Sie sinnvolle Vor- und Nachbedingung der Funktion vor, um damit die Richtigkeit der partition-Funktion entsprechend mit Hilfe von **assert**-Anweisungen überprüfen zu können.

3. Aufgabe (8 Punkte)

Beweisen Sie die Terminierbarkeit der folgenden Programmformel aus unserem 6. Übungsblatt.

$$\{P \equiv m \geq 0 \wedge b > 0\}$$

res = 1

i = 0

$$\{INV \equiv (res == b ** i) \wedge (m \geq i \geq 0) \wedge (b > 0)\}$$

while i < m:

res = res * b

i = i + 1

$$\{Q \equiv res == b ** m\}$$

4. Aufgabe (8 Punkte)

Folgendes Python-Programm berechnet ohne Multiplikation und Division die kleinste ganze Zahl, die größer oder gleich der Wurzel einer eingegebenen ganzen Zahl n ist.

Es wird für diesen Algorithmus die Gültigkeit folgender Gleichung verwendet:

$$n^2 = S(n) + S(n-1) \quad \text{mit} \quad S(n) = \sum_{i=0}^n i = \frac{1}{2} n(n+1)$$

```
n = int(input("n=")) # die Eingabe wird nicht berücksichtigt
```

```
{P ≡ n > 0}
```

```
a = 0
```

```
k = 1
```

```
{INV ≡ a == S(k-1) and S(k-1) + S(k-2) < n }
```

```
while a + a + k < n:
```

```
    a = a + k
```

```
    k = k + 1
```

```
{INV and (a + a + k) ≥ n}
```

```
{Q ≡ (k-1)2 < n ≤ k2}
```

- Zeigen Sie das **INV** eine Invariante der **while**-Schleife des Programms ist. Geben Sie dazu die jeweils verwendeten Hoare-Regeln an.
- Beweisen Sie die Korrektheit des Programms. Das bedeutet, Sie müssen die partielle Korrektheit und die Terminierbarkeit des Programms beweisen.