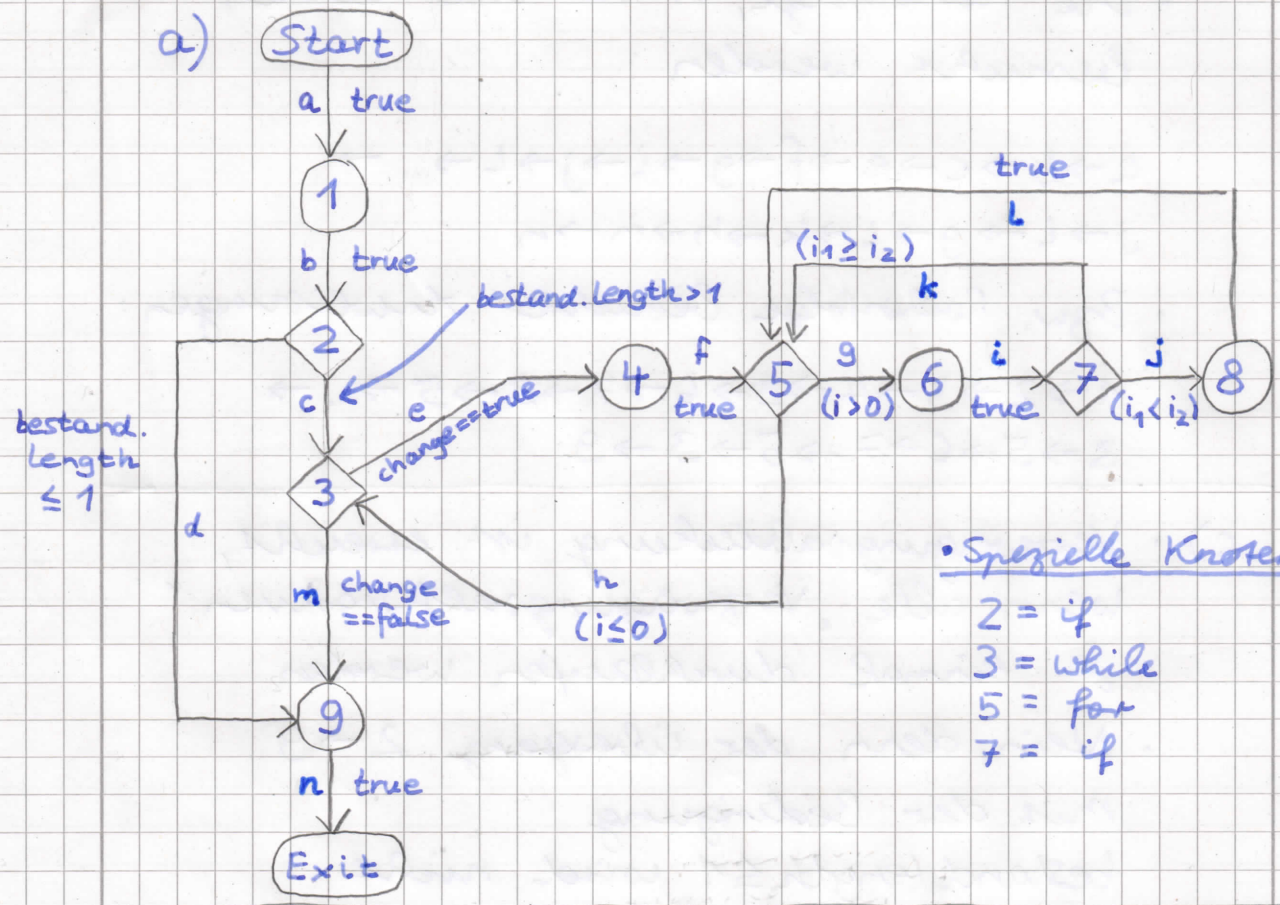
09.01.20 | SWT | ü12 (5)

Praktische Aufgaben

Tobias Ctistis, Gruppe 4

Aufgabe 1)

a)



- Spezielle Knoten:
  - 2 = if
  - 3 = while
  - 5 = for
  - 7 = if

```
public int[] sortiere(int[] bestand) {                    //
Anweisungsnr.
  (1) boolean change = true;                              //  1
  (2) if (bestand.length > 1) {                           //  2
        (3) while (change) {                              //  3
              (4) change = false;                         //  4
              (5) for (int i = bestand.length - 1;        //  5
                       i > 0;                             //  6
                       i--) {                             //  7
                    (6) { int i1 = bestand[i];            //  8
                          int i2 = bestand[i - 1];        //  9
                       (7) if (i1 < i2) {                 // 10
                             (8) { bestand[i] = i2;       // 11
                                   bestand[i - 1] = i1;   // 12
                                   change = true;         // 13
                             }
                       }
              }
        }
  (9) return bestand;                                     // 14
}
```

b) • Falschherum sortiertes Eingabearray, bspw.: 5 | 4 | 3 | 2 | 1

• Die Reihenfolge, in der die Kanten besucht werden:

a → b → c → e → f → g → i → j → L → ... →
j → L → g → i → k → h → m → n

• Bzw. Reihenfolge besuchter Anweisungen:

1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 5 → ... →
8 → 5 → 6 → 7 → 5 → 3 → 9

---

c) • Verzweigungsabdeckung ist erreicht, wenn alle „Verzweigungsalternativen" je einmal durchlaufen werden

• Nein, denn der Übergang 2 → 9 mit der Bedingung bestand.length ≤ 1 wird nicht durchlaufen

• Füge neuen Testfall hinzu: Leeres Eingabearray

• Reihenfolge der Kanten:

a → b → d → n

• Reihenfolge der Anweisungen:

1 → 2 → 9

---

d) • Pfade ohne mehrfache Schleifendurchläufe: Sortiertes Eingabearray: mit 2 Elementen

- Reihenfolge: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 5 → 3 → 9
- Algorithmus erkennt, dass sortiert und checkt nur 2 Elemente

---

* ist Pfad, der alle Vergleiche mit dem Ergebnis ausführt, dass nicht getauscht wurde (immer change = false)

- Leeres Eingabearray
  * Kein Schleifendurchlauf
  * Übergänge: 1 -> 2 -> 9
  * Es wird erkannt, dass nichts zu tun ist und direkt beendet

- Mögliche Pfade sind also:
  * Leeres Eingabearray
  * Sortiertes Eingabearray mit 2 Elementen (da nur Pfade ohne mehrfache Schleifen-durchläufe betrachtet werden)

- Ansonsten abhängig vom Eingabearray
  eventuell unendlich viele Pfade
  => Standardmäßig hier 0-2ten Durchlauf pro Schleife testen
  => D.h. getestet werden muss zusätzlich:
  * Falschherum sortiertes Array (In jedem außer dem letzten Vergleich getauscht)
  * Für die for-Schleife (5) stellt die if-Bedingung (2) in jedem Fall zwei Durchläufe sicher, da die Laufvariable i > 1 sein muss

=> Insgesamt 3 zu testende Pfade

Aufgabe 2)

a) - Äquivalenzklassen:

Jeweils für Tag, Monat:
einstellig (jeweils mit / ohne führender Null),
zweistellig (-> Bezug auf 21. Jahrhundert)

Jahr: leer, ungültig (mit Buchstaben etc.),
zweistellig, vierstellig, Stelligkeit: 3, >4
Monat: leer, ungültig (mit Buchstaben etc.),
<1, 1<x<12, >12
Tag: leer, ungültig (mit Buchstaben etc.),
<1, 1<x<Monatsletzter Tag, >Monatsletzter Tag

- Z. B. :

```
aa-01-18    //Fehler: Buchstabe     //NULL
12-12-12    //Korrekt               //12.12.2012 12:00
12-35-12    //Korrekt               //12.12.2012 12:00
35-12-12    //Korrekt               //31.12.2012 12:00
3-2-2020    //Korrekt               //03.02.2020 12:00
```

c) - Zweistelliges Jahr mit Werten >= 10