

Aufgabe 4: Datenflussorientiertes Testen

Gegeben Sei folgende Java-Methode `sort` zum Sortieren eines Feldes ganzer Zahlen:

```
4   public static int[] sort(int[] array) {
5       boolean swapped;
6       int swapTmp;
7       int[] newArray = (int[]) array.clone();
8       do {
9           swapped = false;
10          for (int index = 0; index < newArray.length - 1; index++) {
11              if (newArray[index] > newArray[index + 1]) {
12                  swapTmp = newArray[index];
13                  newArray[index] = newArray[index + 1];
14                  newArray[index + 1] = swapTmp;
15                  swapped = true;
16              }
17          }
18      } while (swapped);
19      return newArray;
20  }
21 }
```

- (a) Konstruieren Sie den Kontrollflussgraphen des obigen Code-Fragments und annotieren Sie an den Knoten und Kanten die zugehörigen Datenflussinformationen (Definitionen bzw. berechnende oder prädikative Verwendung von Variablen).
- (b) Nennen Sie die maximale Anzahl linear unabhängiger Programmpfade, also die zyklomatische Komplexität nach McCabe.
- (c) Geben Sie einen möglichst kleinen Testdatensatz an, der eine 100%-ige Verzweigungsüberdeckung dieses Moduls erzielt.
- (d) Beschreiben Sie kurz, welche Eigenschaften eine Testfallmenge allgemein haben muss, damit das datenflussorientierte Überdeckungskriterium „all-uses“ erfüllt.