

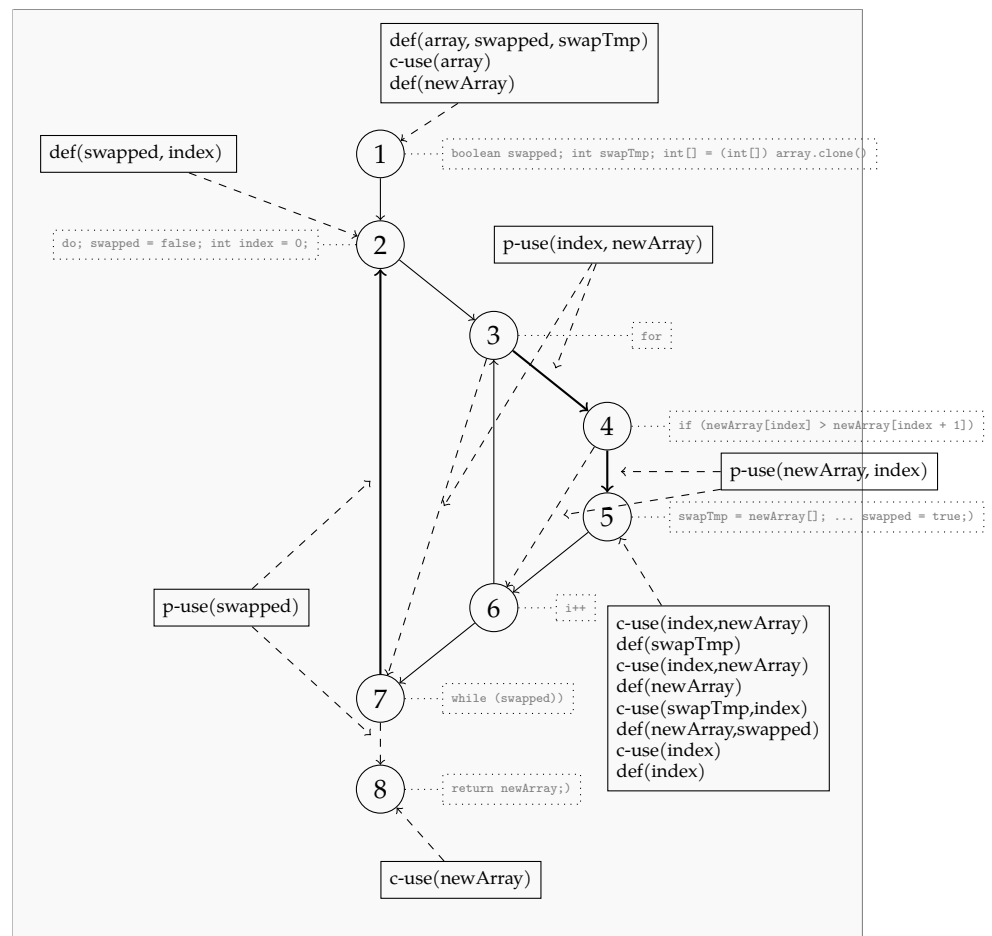
## Aufgabe 3

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 }
```

[github: raw](#)

- (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.

Der Graph hat 8 Knoten und 10 Kanten. Daher ist die zyklomatische Komplexität nach McCabe gegeben durch  $10 - 8 + 2 = 4$ .

- (c) Geben Sie einen möglichst kleinen Testdatensatz an, der eine 100%-ige Verzweigungsüberdeckung dieses Moduls erzielt.

Die Eingabe muss mindestens ein Feld der Länge 3 sein. Ansonsten wäre das Feld schon sortiert bzw. bräuchte nur eine Vertauschung und die innere if-Bedingung wäre nicht zu 100% überdeckt. Daher wählt man beispielsweise `array = [1,3,2]`.

- (d) Beschreiben Sie kurz, welche Eigenschaften eine Testfallmenge allgemein haben muss, damit das datenflussorientierte Überdeckungskriterium „all-uses“ erfüllt.

Das Kriterium all-uses ist das Hauptkriterium des datenflussorientierten Testens, denn es testet den kompletten prädikativen und berechnenden Datenfluss. Konkret: von jedem Knoten mit einem globalen  $\text{def}(x)$  einer Variable  $x$  existiert ein definitions-freier Pfad bzgl.  $x$  ( $\text{def-clear}(x)$ ) zu jedem erreichbaren Knoten mit einem  $\text{c-use}(x)$  oder  $\text{p-use}(x)$ .