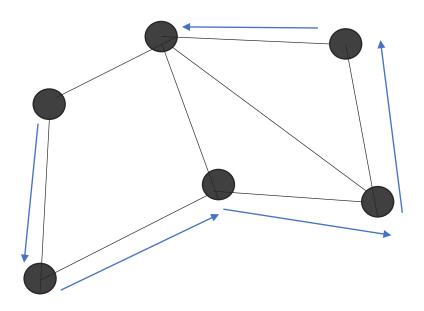
A09 Tiefensuche: Zusammenhängender Graph

private void **depthFirstSearch** (Graph g, int node, boolean[] vis)

Zuerst wird kontrolliert, ob der node bereits besucht wurde. Nun werden per foreach alle gewichteten Kanten w von den Graphen g abgegangen und jeweils rekursiv die depthFirstSearch von Graphen g, dem nächstfolgenden Knoten und dem booleschen Array mit den besuchten Knoten aufgerufen Laufzeit: O (|V| + |E|) V = Anzahl der Knoten E = Anzahl der Kanten



public void **getNumberOfComponents** (Graph g)

AnzahlK gibt die Anzahl der Kanten an und wird erstmal mit 0 initialisiert. Der SummeN wird bereits die Anzahl der Knoten des Graphen g zugeordnet.

Es wird ein neues boolesche Array der Länge der Knoten angelegt, um alle besuchten zu speichern.

Falls SummeN 0 oder 1 ist, wird sofort diese Zahl retourniert und man muss die for-Schleife gar nicht betreten.

Nun werden alle Kanten abgegangen, sofern an der Stelle i im Array der besuchten Knoten, noch ein Eintrag enthalten ist. Hierfür wird depthFirstSearch aufgerufen und die Anzahl der Kanten erhöht (anzahlK ++)

Am Ende wird die Anzahl der Kanten retourniert.