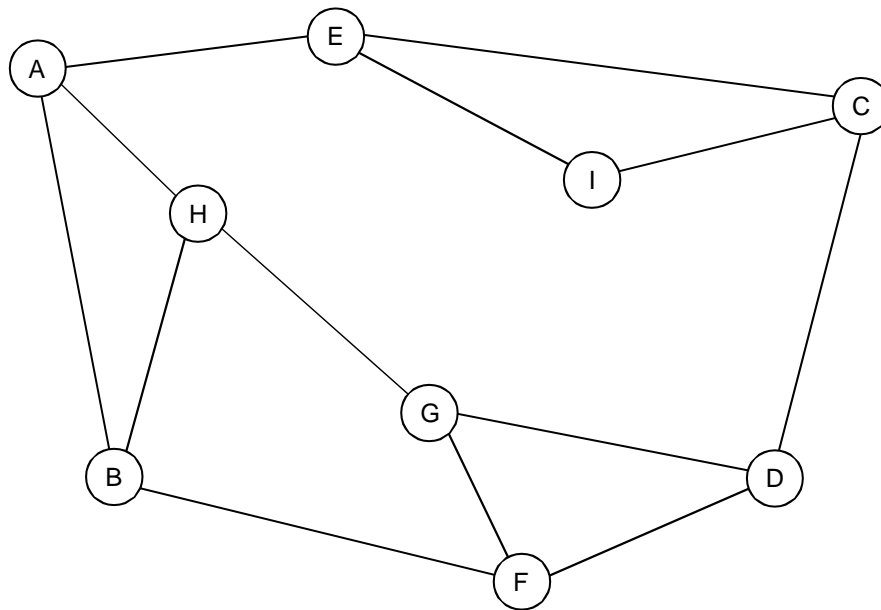
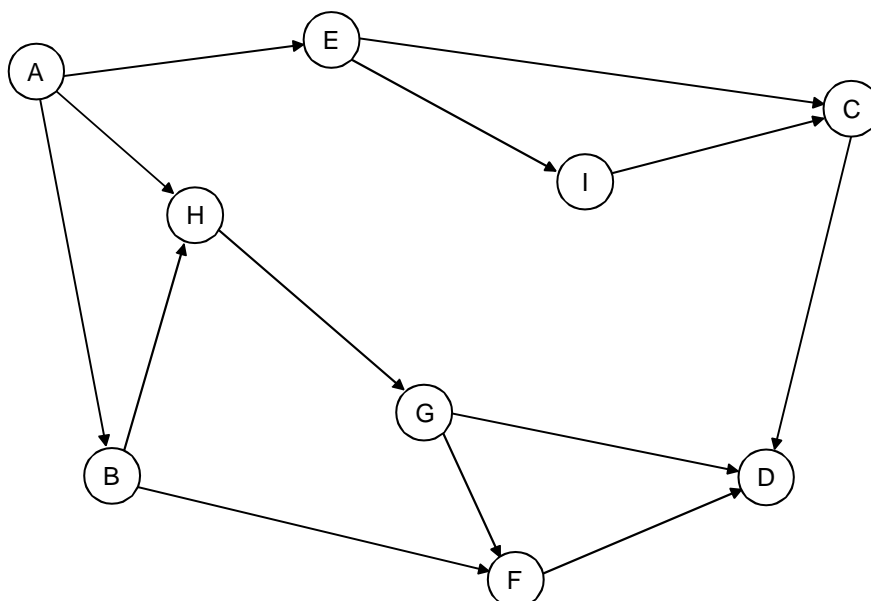


VL12, Aufgabe 1 (Übung)

- Zeichnen Sie die Adjazenzlistenstruktur für untenstehenden Graphen. Die Nachbarknoten eines Knotens sollen in alphabetischer Reihenfolge in die Listenstruktur eingetragen werden.
- Führen Sie auf Basis dieser Listenstruktur eine Breitensuche beginnend bei Knoten A durch. Wenn ein Knoten mehrere Nachbarn hat, sollen die Nachbarn in alphabetischer Reihenfolge durchlaufen werden.
- Führen Sie auf Basis der Listenstruktur aus a) eine Tiefensuche beginnend bei Knoten A durch. Verwenden Sie für die Auswahl der Nachbarn die Reihenfolge, die die Methode `rekDfs` im Beispielprogramm zur Vorlesung benutzt.

**VL12, Aufgabe 2 (Übung)**

Der Graph aus der vorigen Aufgabe wurde in einen gerichteten Graphen umgewandelt. Sortieren Sie die Knoten topologisch!



VL12, Aufgabe 3 (Übung)

In Aufgabe 4 sollen Sie ein Programm schreiben, welches die Zusammenhangskomponenten eines Graphen findet und ausgibt. Diese Aufgabe dient als Vorbereitung. Gegeben ist folgender Graph durch seine Adjazenzmatrix:

	A	B	C	D	E	F	G
A		x					
B			x				
C		x					
D							
E				x			
F		x	x				
G							

Zeichnen Sie den Graphen. Welche Zusammenhangskomponenten gibt es? Überlegen Sie, wie man die Zusammenhangskomponenten eines Graphen, der durch seine Adjazenzmatrix gegeben ist, bestimmen kann. Notieren Sie Ihren Algorithmus in Pseudocode. Prüfen Sie Ihren Algorithmus anhand des Beispielgraphen.

VL12, Aufgabe 4 (Praktikum)

Laden Sie das ZIP-Archiv `UEB12.zip` herunter. Implementieren Sie in `Graph.java` den in Aufgabe 3 entwickelten Algorithmus zur Bestimmung der Zusammenhangskomponenten. Betrachten Sie dazu auch die Beispielprogramme zur Vorlesung.

Beginnen Sie mit der Methode `isKante`, und berücksichtigen Sie bereits dort, dass in gerichteten Graphen zur Bestimmung der Zusammenhangskomponenten **ausnahmsweise** die Kanten als ungerichtet gelten (ähnlich wie Feuerwehr und Polizei, die im Notfall die Richtung von Einbahnstraßen ignorieren dürfen).