

Graphen

Ein Graph besteht aus einer Menge von Knoten und einer Menge von Kanten. Durch Kanten werden jeweils zwei Knoten miteinander verbunden; diese sind dann benachbart oder Nachbarknoten. Bei gerichteten Kanten unterscheidet man bei den Nachbarknoten zwischen Vorgänger- und Nachfolgerknoten. Bei bewerteten Graphen wird den Kanten ein Gewicht zugeordnet. Insbesondere sind Bäume und Listen spezielle Graphen.

Wege durch Graphen

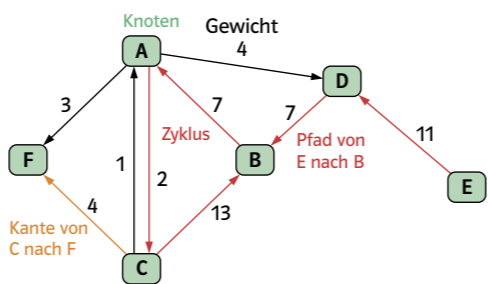
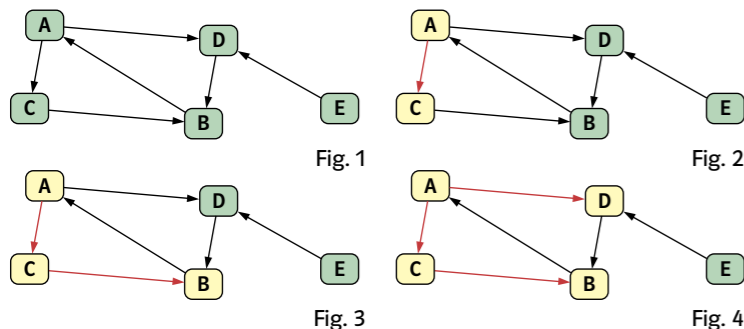
Eine Folge von aufeinanderfolgenden Kanten stellt einen Pfad dar. Gibt es einen Pfad von einem Knoten zu einem zweiten Knoten, so ist der zweite vom ersten aus erreichbar. In ungerichteten zusammenhängenden Graphen ist jeder Knoten von jedem anderen aus erreichbar. Ein Zyklus ist ein Pfad, der in dem Knoten endet, von dem aus er begonnen hat.

Repräsentation von Graphen

Einen Graphen kann man mithilfe einer Adjazenzmatrix repräsentieren. Bei unbewerteten Graphen lauten die Einträge dabei „wahr“ oder „falsch“, je nachdem, ob eine Kante zwischen den entsprechenden Knoten vorhanden ist oder nicht. Im Falle von bewerteten Graphen werden die Kantengewichte in die Zellen der Matrix eingetragen.

Suche in Graphen

Es gibt verschiedene Verfahren, alle Knoten eines Graphen nacheinander zu bearbeiten. Bei der Tiefensuche werden ausgehend von einem Startknoten *s* alle von *s* aus erreichbaren Knoten untersucht. Dabei wird zunächst in die Tiefe gegangen, d.h., es wird ein Nachbar- bzw. Nachfolgerknoten eines bereits erreichten Knotens weiter verfolgt, bevor die anderen Nachbarn bzw. Nachfolger dieses Knotens besucht werden. Bei der Umsetzung des Tiefensuche-Algorithmus sind die Knoten während der Bearbeitung geeignet zu markieren, um zu verhindern, dass man in einen Zyklus gerät. Startet man bei dem Graphen aus Fig. 1 mit dem Knoten A, so ist eine Tiefensuche-Abarbeitung der Knoten in der Reihenfolge A, C, B und D möglich (Fig. 2, 3 und 4). Der Knoten E wird nicht besucht, da er vom Startknoten A aus nicht erreichbar ist.



Die Abbildung zeigt einen bewerteten, gerichteten Graphen mit Zyklen. Dieser wird durch folgende Adjazenzmatrix repräsentiert:

	A	B	C	D	E	F
A			2	4		3
B	7					
C	1	13				4
D		7				
E				11		
F						

Beispielsweise zeigt der Eintrag 7 in der Zelle der Zeile B und der Spalte A an, dass es von B nach A eine Kante mit dem Gewicht 7 gibt.

```

class Graph {
    private Knoten[] knotenliste;
    private int[][] adjazenzmatrix;
    private int anzahl;
    ...
    public void tiefensuche(int startIndex) {
        // Initialisierung
        tiefensucheKnoten(startIndex);
    }
    private void tiefensucheKnoten(int vIndex) {
        // Knoten mit vIndex als besucht
        // markieren
        ...
        for (int i = 0; i < anzahl; i++) {
            if (adjazenzmatrix[vIndex][i]
                &&!knotenliste[i].markierungGeben()) {
                tiefensucheKnoten(i);
            } // Ende if
        } // Ende for
    }
}

class Knoten {
    private Datenelement inhalt;
    private boolean markierung;
    ...
}
    
```

1 Graphen der Restaurantketten

Fig. 1 zeigt das Datenbankmodell, wie es für größere Restaurantketten typisch ist. a) Welche Unterschiede bestehen zwischen diesem Klassendiagramm und dem im Lehrtext diskutierten Graphen? Kann ein Klassendiagramm als Graph angesehen werden? Wenn ja, ist dieser gerichtet? b) In Band 3 dieser Lehrbuchreihe wurden Strategien für die systematische Implementierung von Klassendiagrammen wie in Fig. 1 präsentiert. Vergleichen Sie die dort vorgeschlagene Implementierungstechnik mittels Assoziationsklassen mit der in diesem Kapitel eingesetzten Technik der Adjazenzmatrizen. c) Wie könnten die Kanten dabei implementiert werden? Weshalb kann es in sehr vielen Fällen hilfreich sein, bei der Implementierung von Klassendiagrammen wie in Fig. 1 von einem ungerichteten Graphen auszugehen?

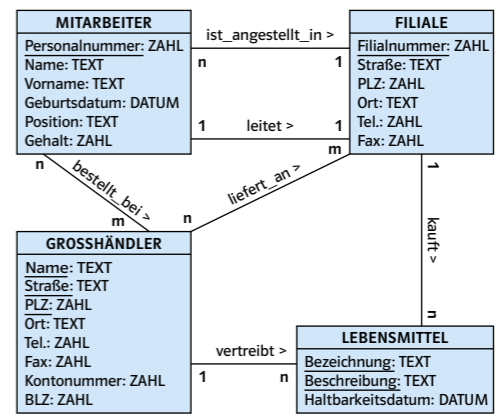


Fig. 1

2 Hektisches Skifahren

Fig. 2 zeigt einen Ausschnitt des Pistenplans des Skigebiets Sudelfeld-Bayrischzell. Peter Eilig hat wenig Zeit, möchte das Skigebiet jedoch möglichst komplett kennenlernen und gleichzeitig einen Lift genau einmal benutzen. a) Als systematischer Mensch erstellt er sich am Abend vor dem Pistentag einen abstrakten Graphen, der Pisten und Lifte beschreibt. Wie könnte dieser aussehen? b) Welche Eigenschaften besitzt der Graph, der das Pistenetz beschreibt? c) Geben Sie die zugehörige Adjazenzmatrix an. d) Wie müsste Peter vorgehen, wenn er seinen „Fahrplan“ unter Verwendung von Tiefensuche finden müsste? Versuchen Sie intuitiv und ohne Verwendung der Tiefensuche, eine pragmatische Lösung des Problems zu finden, bei der die Mehrfachbenutzung von Liften möglichst selten sein soll.



Fig. 2

3 Tiefensuche

a) Führen Sie den Tiefensuche-Algorithmus auf Papier am Graphen aus Fig. 3 durch. Beginnen Sie Ihre Suche beim Knoten A. b) Führen Sie den Tiefensuche-Algorithmus am Graphen der Fig. 3 durch und beginnen Sie Ihre Suche nun beim Knoten D. Welcher wesentliche Unterschied besteht im Vergleich zum Durchlauf von Teilaufgabe a)? c) Im Rahmen einer Implementierung kann in diesem Fall vereinfachend char statt DATENELEMENT als Typ des Attributes inhalt gewählt werden. Erstellen Sie eine entsprechende Implementierung. Testen Sie Ihre Implementierung anhand des Graphen aus Fig. 3.

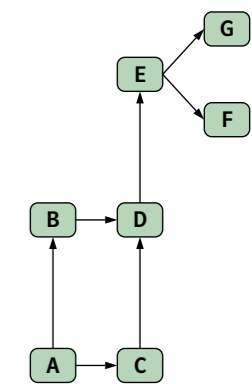


Fig. 3