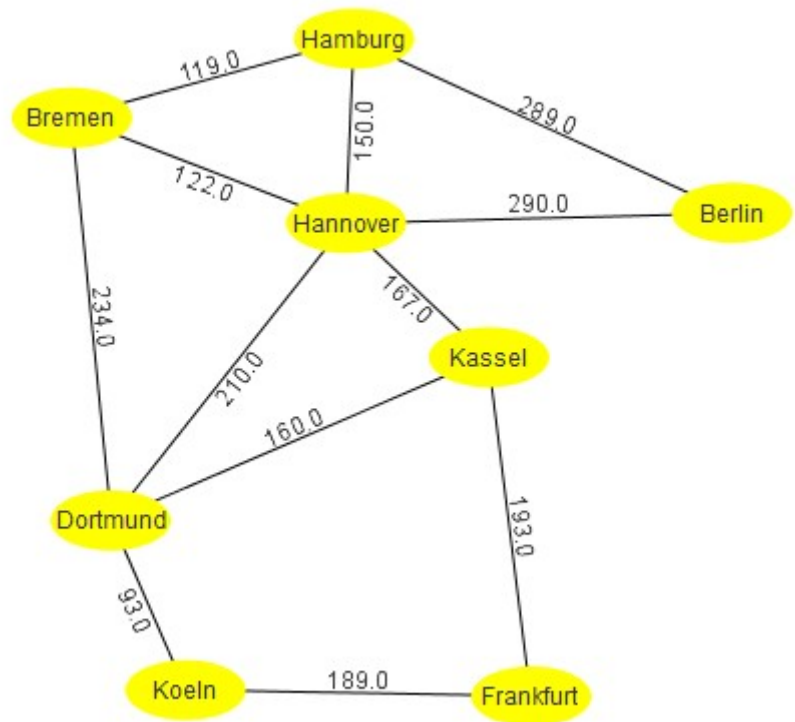


Graph

Ein Graph ist eine Struktur, die eine Menge von Objekten zusammen mit den zwischen diesen Objekten bestehenden Verbindungen repräsentiert.

Die mathematischen Abstraktionen der Objekte werden dabei **Knoten** des Graphen genannt. Die paarweisen Verbindungen zwischen Knoten heißen **Kanten**. Die Kanten können Zahlen, sogenannte **Gewichte** haben.

Ein Graph kann entweder als Graph, als **Adjazenzliste** oder als **Adjazenzmatrix** dargestellt werden.



Adjazenzmatrix

Knoten und Kanten eines Graphen können in Form einer Matrix dargestellt werden. Die Matrix ist dabei spiegelsymmetrisch.

Der oben dargestellte Graph hat folgende **Adjazenzmatrix**:

	Berlin	Bremen	Dortmund	Frankfurt	Hamburg	Hannover	Kassel	Koeln
Berlin					289	290		
Bremen			234		119	122		
Dortmund		234				210	160	93
Frankfurt							193	189
Hamburg	289	119				150		
Hannover	290	122	210		150		167	
Kassel			160	193		167		
Koeln			93	189				

Traversierungen von Graphen

Tiefendurchlauf

Beim Tiefendurchlauf (engl. Depth First Search - DFS) durch einen Baum nimmt man ausgehend von einem Startknoten den ersten Nachbarknoten. Von diesem nimmt man wieder den ersten Nachbarknoten usw. Wenn man dann in eine Sackgasse gerät, geht man eine Stufe zurück und nimmt den nächsten Nachbarknoten. Natürlich werden Knoten, die man schon besucht hat, nicht noch einmal berücksichtigt.

Der Tiefendurchlauf entspricht der Preorder-Traversierung eines Baumes.

Beispiel:

Es gibt für jeden Startknoten mehrere mögliche Tiefendurchläufe, denn man kann sich bei den Nachbarknoten frei entscheiden, welchen man zuerst nimmt. In diesem Beispiel werden die Nachbarknoten immer nach alphabetischer Ordnung genommen.

Tiefendurchlauf für den Startknoten Frankfurt:

Zu Anfang kann man immer direkt weitergehen:

Frankfurt -> Kassel -> Dortmund -> Bremen -> Hamburg -> Berlin -> Hannover

Von Hannover aus ist kein freier Knoten mehr erreichbar. Deswegen muss man jetzt in der Liste zurückgehen, bis man zu einem Knoten kommt, der noch einen freien Nachbarknoten hat. Das ist in diesem Fall Dortmund (der freie Nachbarknoten ist Koeln). Das heißt, Koeln wird als nächstes angehängt, und man würde von Köln aus weitersuchen (wenn es noch freie Knoten gäbe...)

Ergebnis:

Frankfurt -> Kassel -> Dortmund -> Bremen -> Hamburg -> Berlin -> Hannover -> Koeln

Breitendurchlauf

Der Breitendurchlauf (engl. breadth first search - BFS) ist eine Methode, um alle Knoten eines Graphen aufzuzählen.

Der Breitendurchlauf in einem Binärbaum ist genau Levelorder

Mit dem Breitendurchlauf werden die Knoten in folgender Reihenfolge aufgezählt:

- zuerst der Startknoten,
- dann die Nachbarknoten des Startknotens, d.h. alle Knoten, die vom Startknoten aus über eine Kante erreichbar sind.
- dann die Knoten, die vom Startknoten aus mit zwei Kanten erreichbar sind.
- usw.

Knoten, die schon einmal aufgezählt wurden, werden natürlich nicht wieder aufgezählt.

Im Binärbaum ist der Breitendurchlauf genau Levelorder.

Beispiel:

Beim Breitendurchlauf gibt es für jeden Startknoten mehrere Möglichkeiten, denn man kann zwischen den Nachbarknoten wählen. Hier werden die Nachbarknoten immer in alphabetischer Reihenfolge betrachtet.

Breitendurchlauf für den Startknoten Frankfurt

Erst der Startknoten und seine Nachbarknoten:

- Frankfurt -> Kassel -> Koeln

Jetzt wird von den Nachbarknoten der erste genommen und dessen Nachbarknoten werden betrachtet:

- Frankfurt -> Kassel -> Koeln -> Dortmund -> Hannover

Der nächste Knoten in der Liste, der noch freie Nachbarknoten hat, ist Dortmund:

- Frankfurt -> Kassel -> Koeln -> Dortmund -> Hannover -> Bremen

Schließlich die Nachbarknoten von Hannover:

- Frankfurt -> Kassel -> Koeln -> Dortmund -> Hannover -> Bremen -> Berlin -> Hamburg

Beim Breitendurchlauf wird also zuerst die "nähere Umgebung" betrachtet.