Repräsentation von Graphen

Problemstellung 1

Ein Soziogramm (graphische Darstellung der Beziehungen in einer Gruppe) soll aus den Handykontaktdaten erstellt werden. In nebenstehender Liste sieht man, wer welche Nummern gespeichert hat.

Kennt Handynummer:

Adriana kennt Clara, Betül

Betül kennt Clara, Dominik

Clara kennt Adriana

Dominik kennt Betül, Clara

Problemstellung 2 (Traveling Salesman Problem)

Deutschlandkarte [gemeinfrei] URL: https://upload.wikimedia.org/wikipedia/commons/5/57/Karte_Deutschland.png (abgerufen: 3.4.2018)

Ein Geschäftsmann aus Frankfurt muss Kunden in Berlin, Nürnberg und München beraten. Er möchte seine Rundreise so planen, dass er jeden Ort nur genau einmal besucht und die Gesamtfahrstrecke dabei möglichst klein bleibt. Dazu hat er die Entfernungen (in km) in einer Tabelle aufgeschrieben:

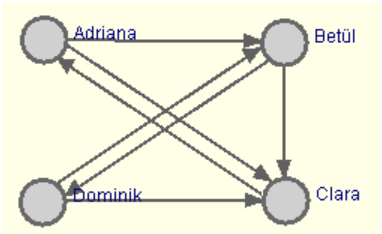
	München	Nürnberg	Frankfurt	Berlin
München	0	170	400	590
Nürnberg	170	0	230	440
Frankfurt	400	230	0	550
Berlin	590	440	550	0

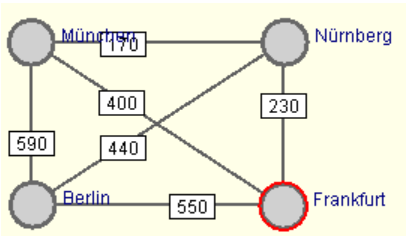
Aufgaben:

1. *Entscheide in beiden Situationen, ob es sich um einen gerichteten oder einen ungerichteten Graphen handelt. Entscheide in beiden Situationen, ob es sich um gewichteten oder einen ungewichteten Graphen handelt.*

   Problemstellung 1: gerichteter, ungewichteter Graph
   Problemstellung 2: ungerichteter, gewichteter Graph

2. *Stelle die Graphen beider Problemstellungen dar.*





3. *Erläutere, wie man einen gewichteten Graphen in einer Adjazenzliste speichern könnte. Beschreibe, wie die Adjanzenzliste aussehen würde, wenn der Graph ungerichtet wäre.*

   In der Adjazenzliste müsste hinter jedem Knoten die Gewichtung der Kante zu dem Knoten angegeben werden. Wenn der Knoten k2 in der Liste des Knoten k1 steht, muss bei einem ungerichteten Graphen automatisch auch k1 in der Liste von k2 auftauchen.

4. *Erläutere, wie man einen ungewichteten Graphen in einer Adjazenzmatrix speichern*

kann. Beschreibe, wie die Adjazenzmatrix aussähe, wenn der Graph gerichtet wäre.

In der Adjazenzmatrix muss bei ungewichteten Graphen nur noch festgehalten werden, ob eine Kante vorhanden ist oder nicht. Das kann man z.B. durch eine 1 (Kante vorhanden) oder 0 (keine Kante vorhanden) angeben. Ist der Graph gerichtet, ist die Matrix nicht mehr symmetrisch zur Diagonalen.

5. *Stelle folgende als Adjazenzmatrix oder Adjazenzliste gegebenen Graphen dar.*

a)
$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

b)
$$\begin{pmatrix} 0 & 4 & 1 & 0 & 1 \\ 4 & 0 & 2 & 0 & 0 \\ 1 & 2 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 5 & 1 & 0 \end{pmatrix}$$

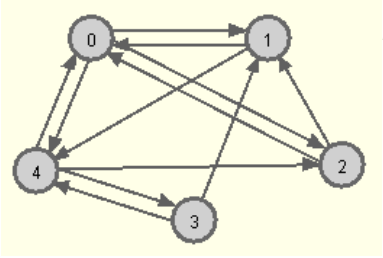
c)

Knoten	Kanten
0	2; 4
1	2; 3; 4
2	0; 1; 4
3	1
4	0; 1; 2

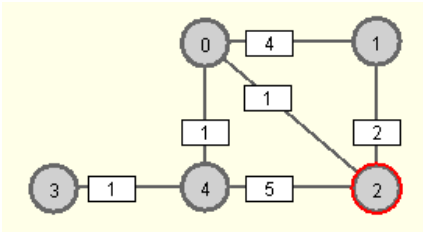
d)

Knoten	Kanten
0	(1; 12); (2; 4)
1	(3; 41)
2	(0; 8); (4; 2)
3	(0; 12)
4	(0; 3); (2; 5)

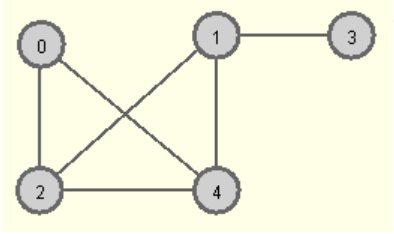
a)



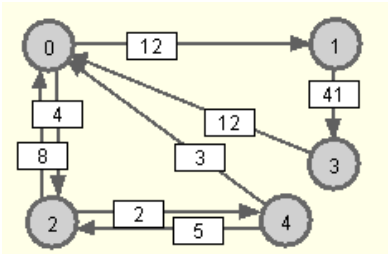
b)



c)



d)



6. *Betrachte die Dateien graph1.csv und graph2.csv im Unterordner beispielgraphen/05_repraesentation des Graphen-Testers in einem Texteditor. Untersuche, wie die Graphen hier gespeichert sind.*

Beide Dateien beginnen mit einigen Basisinformationen über den Graphen, dann kommt der eigentliche Graph. graph1.csv enthält eine Adjazenzliste, graph2.csv eine Adjazenzmatrix. In beiden Formaten ist für jeden Knoten eine Zeile gespeichert. Neben den Kanten des Knoten ist noch die Position des Knotens abgespeichert. Die erste Zahl ist die x-Koordinate, die zweite Zahl die y-Koordinate. Das ist notwendig, um den Graphen zeichnen zu können, für die Algorithmen nicht. graph1.csv ist ein ungewichteter Graph. Daher sind nur die Nummern der Knoten angegeben, wohin die Kanten führen. Bei graph2.csv handelt es sich um einen gewichteten Graph. Daher sind die Matrixeinträge die Gewichte der Kanten.

Effizienzanalyse

Die verschiedenen Repräsentationen der Graphen haben Vor- und Nachteile. Man kann z.B. den benötigten Speicherplatz und die Laufzeit für Zugriffsoperationen auf den Graphen bewerten.

Dabei muss man Anwendungen unterscheiden, bei denen relativ dünne Graphen auftreten, und Anwendungen, bei denen der Graph (nahezu) vollständig ist. Repräsentiert der Graph z.B. eine Karte mit Straßen und Kreuzungen, gehen von jedem Knoten (=Kreuzung) nur wenige Kanten (=Straßen) ab. Mehr als 4-5 Straßen gehen üblicherweise nicht von einer Kreuzung aus. Diese Zahl ändert sich auch nicht, wenn die Anzahl der Knoten steigt. Der Graph ist also sehr dünn. Betrachtet man hingegen das Traveling Salesman Problem, dann kann man von jeder Stadt in jede andere reisen. Der Graph ist also vollständig. Bei einer Erhöhung der der Knotenzahl steigt damit auch die Zahl der Kanten pro Knoten.

Speicherplatz (ungerichteter, gewichteter Graph)

Adjazenzliste: Zwei Zahlen (Nummer des Nachbarknoten, Entfernung) werden gespeichert.
Adjazenzmatrix: Eine Zahl (Entfernung) wird in jeder Zelle gespeichert (0 für keine Kante).

Anzahl der Knoten	dünner Graph (4 Kanten pro Knoten)		vollständiger Graph	
	Adjazenzliste	Adjazenzmatrix	Adjazenzliste	Adjazenzmatrix
5	5·4·2 = 40 Zahlen	5·5 = 25 Zahlen	40 Zahlen	25 Zahlen
6	*48 Zahlen*	*36 Zahlen*	*60 Zahlen*	*36 Zahlen*
7	*56 Zahlen*	*49 Zahlen*	*84 Zahlen*	*49 Zahlen*
8	*64 Zahlen*	*64 Zahlen*	*112 Zahlen*	*64 Zahlen*
9	*72 Zahlen*	*81 Zahlen*	*144 Zahlen*	*81 Zaheln*
n	*O(n) Zahlen*	*O(n²) Zahlen*	*O(n²) Zahlen*	*O(n²) Zahlen*

Ergebnis:

Für dünne Graphen ist die Adjazenzliste weniger speicherplatzintensiv, für (nahezu) vollständige die Adjazenzmatrix.

Laufzeitanalyse

Die Laufzeit einzelner Operationen hängt von der Art der Implementierung ab.

Adjazenzliste: Ein Array enthält für jeden Knoten eine einfach verkettete Liste der Kanten.
Adjazenzmatrix: Zweidimensionales Array.

Operation	dünner Graph (4 Kanten pro Knoten)		vollständiger Graph	
	Adjazenzliste	Adjazenzmatrix	Adjazenzliste	Adjazenzmatrix
Länge einer Kante?	O(1)	O(1)	O(n)	O(1)
Ausgangsgrad eines Knotens?	*O(1)*	*O(n)*	*O(n)*	*O(n)*

Einfügen Kante	$O(1)$	$O(1)$	$O(n)$	$O(1)$
Einfügen Knoten	$O(n)$	$O(n^2)$	$O(n)$	$O(n^2)$
Löschen Kante	$O(n)$	$O(1)$	$O(n)$	$O(1)$
Löschen Knoten	$O(n)$	$O(n^2)$	$O(n)$	$O(n^2)$

Ergebnis:

> *Verschiedene Anwendungsszenarien erfordern die Operationen in unterschiedlicher Häufigkeit. Je nach Bedarf sollte man eine unterschiedliche Repräsentation wählen.*