

### 3.3 Graph mit Adjazenzliste\*

Betrachten wir nochmal den Ansatz, den Graph mit Listen umzusetzen. Letztendlich ist ein Graph nur eine Menge von Knoten und Kanten. Daher können wir beides mit Listen umsetzen.

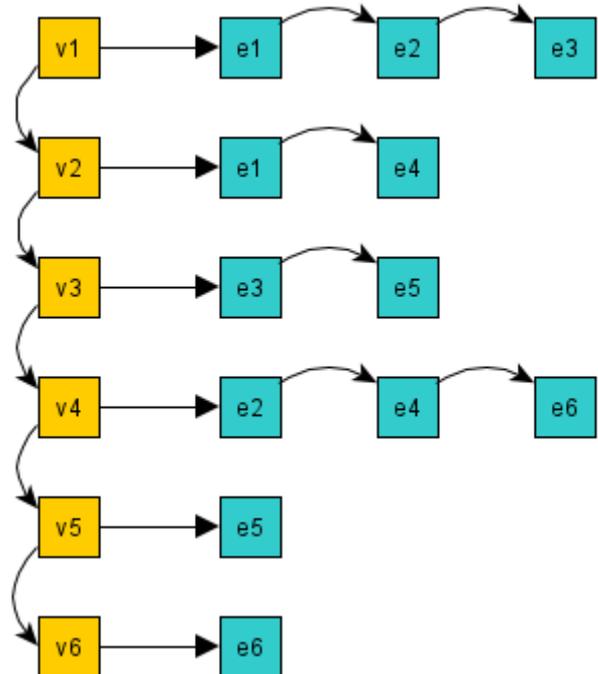
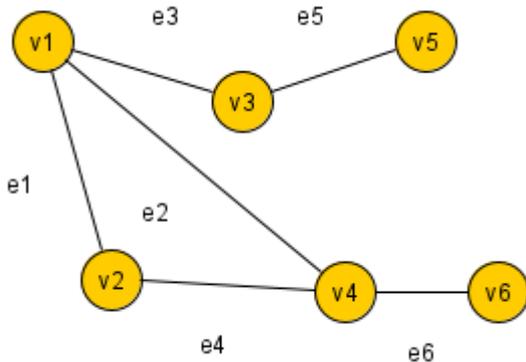
Betrachten wir nochmal ein Beispiel (Abbildung links):

Jeder KNOTEN wird in einer Liste gespeichert. Daraufhin muss jeder KNOTEN eine Liste der KANTEN enthalten, die ihn mit anderen KNOTEN verbindet.

Wenn man sich das veranschaulichen will sieht das so aus wie in der rechten Abbildung. Jeder Knoten bekommt alle Kanten, die ihn mit einem anderen Knoten verbinden.

Wir stellen fest, dass manche Kanten doppelt abgespeichert werden müssen.

z.B. v1 hat als Referenzattribut eine Liste mit e1 als Anfang



Stellt man die beiden Umsetzungen gegenüber stellt man fest, dass beide Umsetzungen ihre Berechtigung haben. Beim Design muss je nach Anwendung entschieden werden.

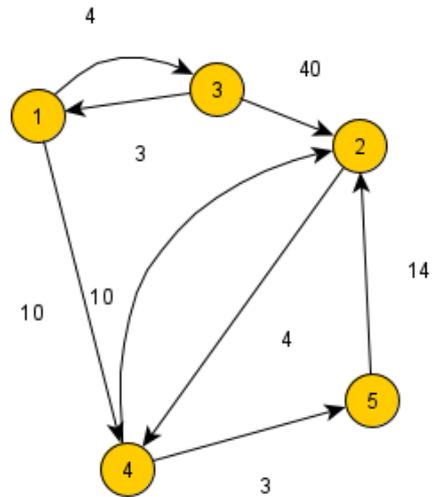
Adjazenzmatrix	Adjazenzliste
<ul style="list-style-type: none"> <li>Kanten sofort erkennbar (vor allem auch für Menschen)</li> </ul>	<ul style="list-style-type: none"> <li>Ob eine Kante existiert muss gegebenenfalls tief in der Liste der Kanten gesucht werden</li> </ul>
<ul style="list-style-type: none"> <li>Bei wenigen Kanten viele „existiert nicht Einträge“ die eigentlich überflüssigen Speicher brauchen</li> </ul>	<ul style="list-style-type: none"> <li>Bei allen ungerichteten Kanten doppelte Speicherung</li> </ul>
<ul style="list-style-type: none"> <li>Leichte Implementierung mit Feld</li> </ul>	<ul style="list-style-type: none"> <li>Komplizierte Implementierung (z.B. Auch Suchen von Kanten)</li> </ul>
<ul style="list-style-type: none"> <li>zum größten Teil statisch</li> </ul>	<ul style="list-style-type: none"> <li>dynamisch</li> </ul>

**Aufgaben:**

1) Überlegen Sie zu den Beispielgraphen und der Adjazenzmatrix (links) des letzten Abb., wie die Adjazenzlisten lauten müssten.

$$\begin{pmatrix} -1 & -1 & 5 & -1 \\ 3 & -1 & 3 & -1 \\ 5 & 5 & 2 & 3 \\ 1 & 2 & 2 & -1 \end{pmatrix}$$

Woran kann man in der Adjazenzliste erkennen, ob ein Graph gerichtet oder zusammenhängend ist?



2) Gegeben ist folgende Adjazenzliste:

Knoten	Kanten des Knoten
A	e1, e2, e3
B	e1, e4
C	e3
D	e2, e4, e5
E	e5

a) Erstellen Sie einen Graph und eine Adjazenzmatrix zur gegebenen Liste.

b) Angenommen man würde den Graph zu einem gerichteten Graph machen wollen, mit e5 geht von D zu E. Wie müsste man die Adjazenzliste ändern, damit dieser neue Graph beschrieben wird?