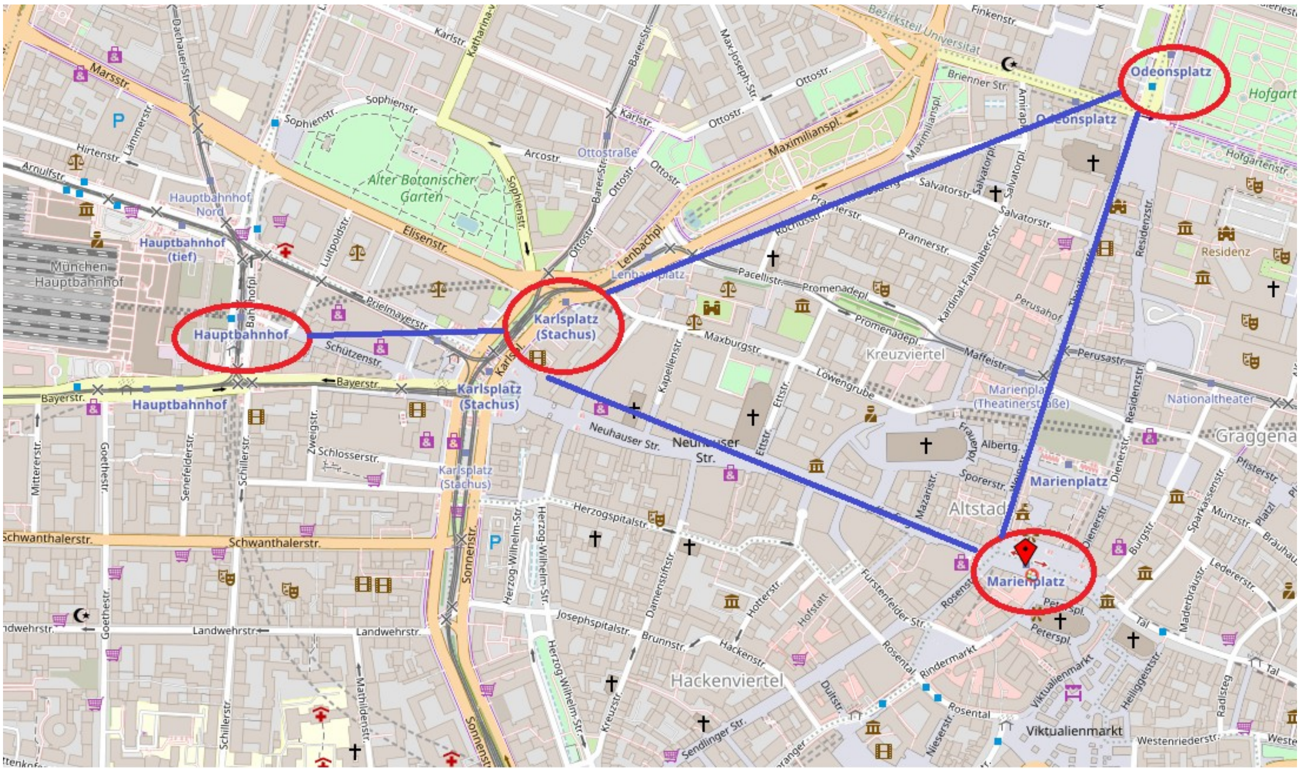
# 3. Graphen

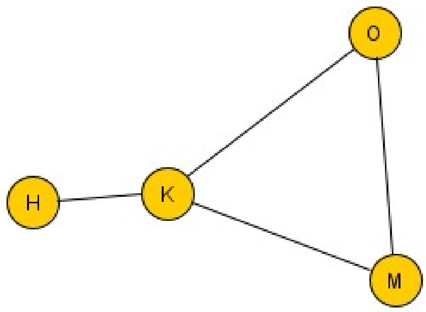
## 3.1 Grundlagen

In der Informatik geht es oft darum, reale Probleme mit Computer entweder zu lösen oder wenigstens das Finden einer Lösung leichter zu machen. Ein alltägliches Problem, für das wir Computer einsetzen, ist das finden eines Weges von einem Ort zu einem anderen. Dabei spielt es keine Rolle, ob wir Wege im Straßenverkehr oder Wege mit dem öffentlichen Personennahverkehr meinen, beide werden heutzutage von Apps für uns übernommen.
Um diese Lösungen zu finden, brauchen wir eine geeignete Modellierung für unser Problem. Betrachten wir also eine Karte von Münchens Straßennetz.



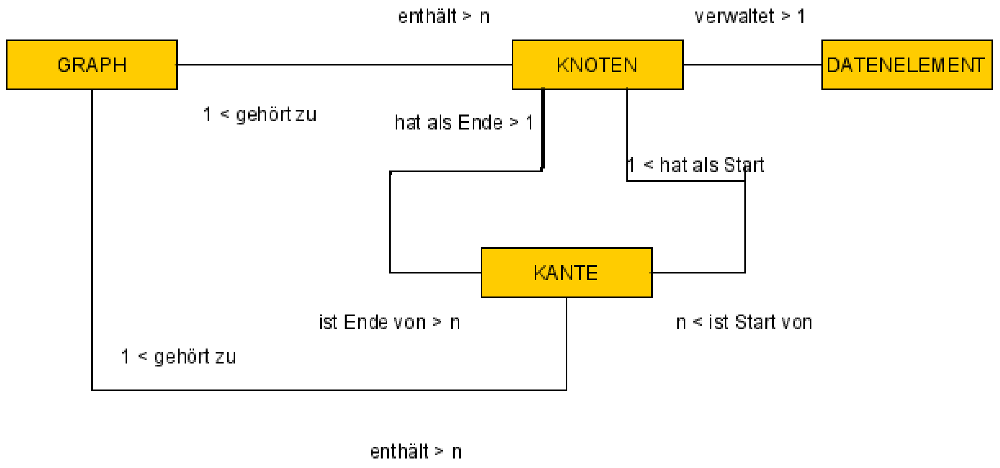

Wir schauen uns vereinfacht mal nur die Verbindung zwischen den Hauptknotenpunkten der Innenstadt Hauptbahnhof, Marienplatz, Karlsplatz und Odeonsplatz an. Es ist beispielsweise sinnvoll, wenn man vom Odeonsplatz zum Hauptbahnhof möchte sinnvoll über den Karlsplatz zu gehen. Wir haben also gewissen KNOTEN, die miteinander verbunden sind.
In der Informatik bezeichnen wir ein solches Gebilde als **GRAPH**. Es ist eine dynamische Datenstruktur, die die Speicherung beliebig vernetzter Daten ermöglicht. Ein Graph besteht aus KNOTEN und KANTEN.

*<u>Unser Beispiel:</u>* „Straßennetz“
Menge der KNOTEN = { Hauptbahnhof (H); Marienplatz (M); Odeonsplatz (O); Karlsplatz (K)}
Menge der KANTEN = {H/K; K/O; O/M; K/M}

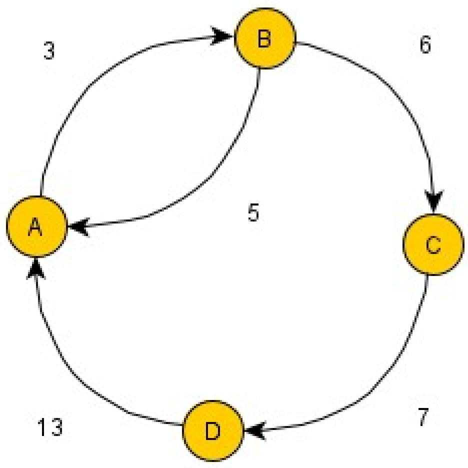



Jeder KNOTEN hat einen Namen und jede KANTE kann ein Gewicht und eine Richtung haben. Hier haben wir einen ungerichteten Graph, der keine Richtungen hat und ohne Gewichtung angegeben ist.

Es ergibt sich folgende erste Modellierung für einen GRAPH:




Im Beispiel Straßennetz würde man eine gerichtete KANTE brauchen, wenn zwei Orte durch eine Einbahnstraße verbunden sind. Eine mögliche Gewichtung könnte die Dauer des Weges sein, was in einer Maps-App durchaus Sinn für die kürzeste Wegfindung ergibt. Manchmal macht es sogar Sinn, dass selbst wenn zwei Orte von beiden Richtungen aus erreichbar sind, trotzdem zwei gerichtete Kanten verwendet werden. In unserem Beispiel könnte ein Weg den Berg rauf gehen und dann bräuchte man mit dem Fahrrad vermutlich länger in die eine Richtung wie in die andere.




Beispiel:
Knoten = {A;B;C;D}
Kanten = {AB; BA; BC; CD; DA}

*Grundbegriffe im Überblick:*

- **gerichtet**: Kanten haben eine Richtung
- **gewichtet:** Jede Kante hat ein Gewicht (eine Zahl) zugeordnet
- **Pfad**: eine Abfolge von Kanten um von einem Knoten zum anderen zu kommen
- **zyklischer Pfad**: ein Pfad, mit Anfang und Ende am gleichen Knoten
- **zusammenhängender Graph**:
  - ungerichteter Graph: Graph, in dem es von jedem Knoten einen Pfad zu jedem anderen Knoten gibt
  - gerichteter Graph: zusammenhängend, wenn der ungerichtete Graph zusammenhängend ist und stark zusammenhängend, wenn es von jedem Knoten zu jedem anderen Knoten einen gerichteten Pfad gibt

**Aufgaben:**

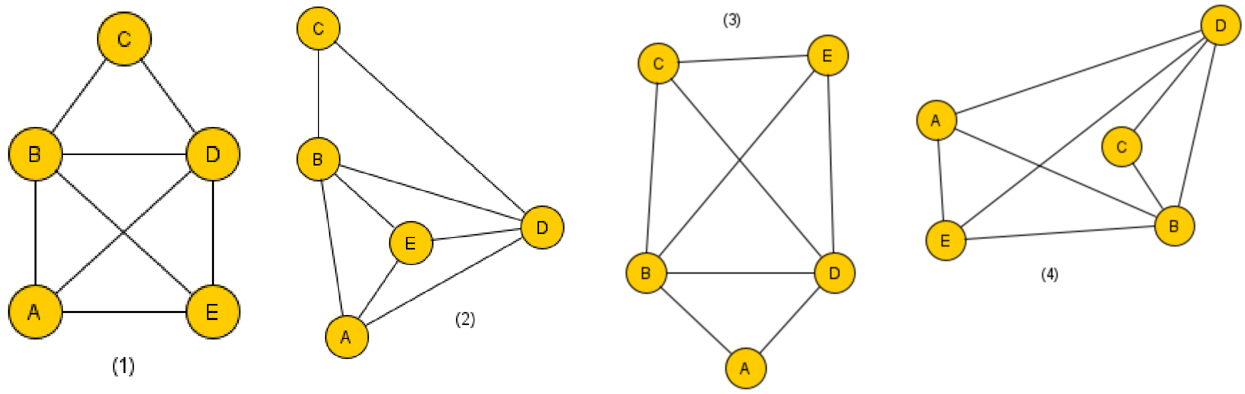
**1)** Informieren Sie sich im Internet über die Flugrouten der Lufthansa. Betrachten Sie die internationalen Flugverbindungen und die nationalen zwischen den Abflugorten ins Ausland.

a) Erstellen Sie einen ungewichteten, ungerichteten Graphen für die Flugrouten. Als Knotenbezeichner verwenden Sie die international gebräuchlichen Abkürzungen. Stellen Sie den Graph in einem Zeichenprogramm dar.
b) Ergänzen Sie den Graphen aus a) zu einem gewichteten Graphen, der die Flugdauer angibt.
c) Warum ist der ungerichtete gewichtete Graph für die Flugrouten ungeeignet? Was ist die Ursache für die unterschiedlichen Flugzeiten? Arbeiten Sie in den Graphen aus b) die Unterschiede für einige Kanten ein, bei denen Hin- und Rückflugzeiten sich um mehr als fünf Minuten unterscheiden.

**2)** Das Königsberger Brückenproblem ist eine mathematische Fragestellung aus dem frühen 18.Jahrhundert, die man wie folgt veranschaulichen kann.
In der Stadt Königsberg gibt es mehrere Inseln, die durch Flüsse getrennt sind. Über die Flüsse führen Brücken. Kann man einen Spaziergang so gestalten, dass man alle Brücken nur genau einmal überquert?

a) Veranschaulichen Sie die Fragestellung mit einem Graph.
b) Wie würde die Frage nach dem Spaziergang im Graph lauten und was ist die Antwort. Begründen Sie ihr Ergebnis.

**3)** Betrachten Sie folgende Graphen. Welche Graphen sind äquivalent zueinander?




**4)** Erstellen Sie ein eigenes Beispiel mit zwei gerichteten Graphen. Lassen Sie ihren Banknachbar herausfinden, ob die Graphen äquivalent sind.

**5)** Überlegen Sie, wie man das Klassendiagramm implementieren könnte. Welche Datenstrukturen kann man wiederverwenden? Programmieren Sie eine rudementäre Version der Klassen.

Kartenquelle: Open StreetMaps via OpenDatabase License