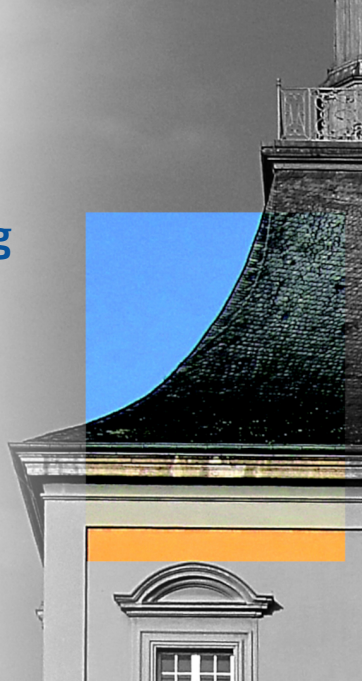
UNIVERSITÄT BONN

Algorithmen und Programmierung

Algorithmen II

Dr. Felix Jonathan Boes
boes@cs.uni-bonn.de
Institut für Informatik



Überblick

UNIVERSITÄT BONN

Bishere und kommende Inhalte

- ✓ Einleitung
- ✓ Motivation: Theorie und Praxis um Sortierverfahren zu studieren
  - ✓ Imperative Programmierung (in C++)
  - ✓ Algorithmen vollständig beschreiben
  - ✓ Speicherverbrauch und Laufzeiten analysieren
- → Motivation: Theorie und Praxis um Graphen zu modellieren
  - Graphen und Algorithmen auf Graphen
  - Abstrakte Datentypen und Datenstrukturen
  - Objektorientierte Programmierung (in C++)

Zur Modellierung im Allgemeinen

Ausblick

Das Modellieren von Systeme und Zuständen ist ein essentieller Bestandteil der Informatik

Das Entwerfen von Modellen ist anspruchsvoll

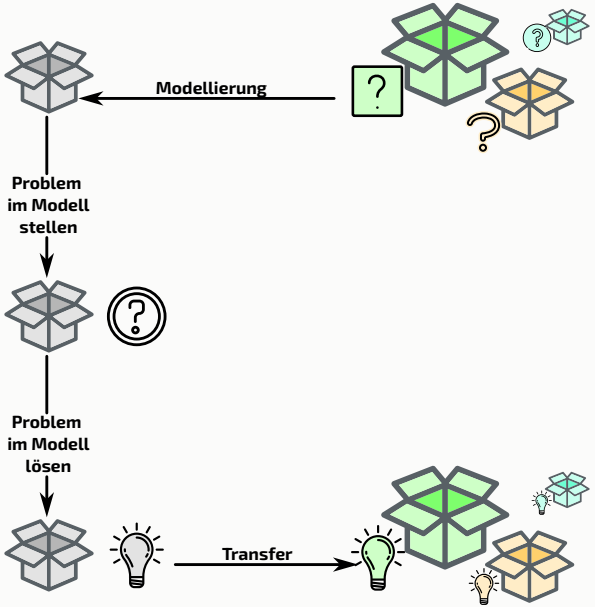
UNIVERSITÄT BONN

Modellierung im Allgemeinen

Sie wollen ein kompliziertes Realweltbeispiel beschreiben und lösen. Zu diesem Zweck wird zunächst ein **Modell** entworfen und das **Problem innerhalb dieses Modells** beschrieben. Dabei ist das Modell so grob wie möglich und gleichzeitig detailliert genug, um alle **wesentlichen Merkmale** des **Szenarios** und des **Problems** innerhalb des Modells ausdrücken zu können.

Dann lösen Sie das Problem innerhalb des Modells und wenden die Lösung des Problems auf Ihr kompliziertes Realweltbeispiel an.

Dieser Abstraktion begegnen wir innerhalb der Naturwissenschaften und der Mathematik durchgehend. Ihnen ist diese Abstraktion von Sortieralgorithmen bereits vertraut.


Modellierung
Problem im Modell stellen
Problem im Modell lösen
Transfer

UNIVERSITÄT BONN

Beispiel Sortieren

Wir haben bereits mehrfach Sortieralgorithmen diskutiert. Im Konkreten wollen wir möglichst schnell sehr konkrete Dinge sortieren.

- 13 Umzugskisten und 7 Jutebeutel nach Gewicht
- 378 korrigierte Erstsemesterklausuren nach Teilnehmernachnamen
- 5261 Wassersorten nach persönlichem Geschmackserlebnis

Ganz natürlich haben wir hier ein Modell vor dem inneren Auge. In diesem Modell studieren wir **vergleichbare Objekte**. Die Objekte können **paarweise miteinander vergleichen** werden und sind sonst eigenschaftslos.

Jeder Sortieralgorithmus der in diesem Modell als effizient eingestuft wird, eignet sich zum Sortieren der oben genannten konkreten Sorterprobleme.

UNIVERSITÄT BONN

Zufallsbekanntschaften

Sie haben heute neue Bekannte geschlossen. Abends Schocken Sie (Würfelspiel) und verlieren erschreckend oft. Sie wollen sich sicher sein, dass Sie nicht schlecht spielen sonder dass Ihre Bekannten gezinkten Würfeln verwenden.

Eine Möglichkeit, um sich sicher zu sein dass die Bekannten sie betrügen, ist es typische Methoden der **analytischen Statistik** zu verwenden. Hier modellieren Sie die Würfel als gleichverteilte Zufallsvariablen. Nun berechnen Sie, wie wahrscheinlich es ist, dass Ihre Bekannten in den vergangenen Spielen bei jedem Wurf mindestens eine 1 würfeln.

Kurze Zeit später entscheiden Sie sich, die neue Bekanntschaft nicht weiter zu vertiefen.

UNIVERSITÄT BONN

Wie gut ist ein Modell?

Es scheint naheliegend sich zu fragen, wie gut **wie gut ein Modell** ist. Diese Frage lässt sich nur schwerlich formal beantworten. Zunächst muss man klären, wie man die **Güte eines Modells misst**. Dazu verwendet man aller Erwartung nach ein (Meta)modell. Hier schließt sich also die Frage an, ob das gewählte Metamodell gut gewählt ist.

Haben Sie Fragen?

Zusammenfassung

Das Modellieren von Systeme und Zuständen ist ein essentieller Bestandteil der Informatik

Das Entwerfen von Modellen ist anspruchsvoll

Graphen

Überblick

Objekte und deren Beziehung werden oft mithilfe von Graphen modelliert

Die Beziehungen können dabei durch weitere Eigenschaften beschrieben werden

Sie lernen erste Graphen und Graphenalgorithmen kennen

UNIVERSITÄT BONN

Was modellieren Graphen?

Mithilfe von **Graphen** modellieren wir **Objekte** und deren (paarweise) **Beziehungen**. Wir betrachten zunächst Beispiele.

- In soziale Netze existieren Personen (Objekte) und paarweise Freundschaften (Beziehungen).
- Beim Entwurf von Computer Chips müssen Bauteile (Objekte) durch Leiterbahnen (Beziehung) miteinander verbunden werden.
- Im Straßennetz existieren Straßen (Beziehung) die an Verzweigungspunkten (Objekte) wie Kreuzungen oder Gabelungen zusammenlaufen.

Je nach Anwendungsfall werden nur Objekte und deren Beziehungen modelliert oder aber es werden den Objekten und Beziehungen noch weitere (für den Anwendungsfall bedeutsame) Zusatzeigenschaften erlaubt.

Graphen

Gerichtete und ungerichtete Graphen

Offene Fragen

Wie werden asymmetrische Beziehungen zwischen Objekten modelliert?

Wie werden symmetrische Beziehungen zwischen Objekten modelliert?

UNIVERSITÄT BONN

Definition gerichteter Graph

Sie begegnen in Ihrem Leben vielen Arten von Graphen. In dieser und folgenden Vorlesungen betrachten wir mehrere Arten von **Graphen ohne Mehrfachkanten**.

(Gerichteter Graph)

Ein **gerichteter Graph** $G = (V, E)$ besteht aus einer endlichen Menge von **Knoten** V und einer (endlichen) Menge von **gerichteten Kanten** $E \subseteq V \times V$.

Wir sagen dass eine Kante $e = (v, w) \in E$ von v nach w verläuft. Außerdem heißt v **Startknoten** und w **Zielknoten**.

UNIVERSITÄT BONN

Beispiel

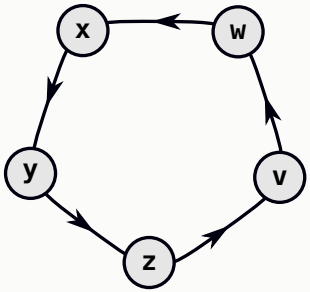
Wir betrachten hier einen gerichteten Graphen $G = (V, E)$ mit fünf Knoten

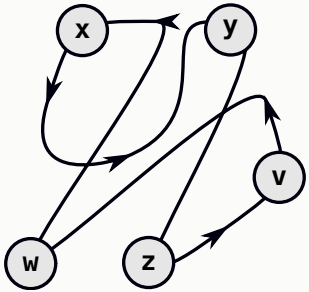
$$V = \{v, w, x, y, z\}$$

und Kanten

$$E = \{(v, w),\ (w, x),\ (x, y),\ (y, z),\ (z, v)\} \subseteq V \times V$$

Hier sind zwei Möglichkeit den Graphen G bildlich zu beschreiben.





Weg

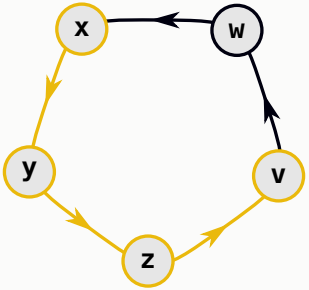
Eine Folge von Kanten $e_1, e_2, \ldots, e_j = (v_j, w_j), \ldots, e_n$ heißt **Weg**, wenn bei jeder Kante der Zielknoten wiederum der Startknoten der Folgekante ist. In anderen Worten ist das genau dann der Fall, wenn $w_i = v_{i+1}$ für $i = 1, \ldots, n-1$ gilt.

Die Länge eines Wegs $e_1, e_2, \ldots, e_n$ ist n. Also hat ein Weg der Länge $n \geq 1$ genau n Kanten und durchläuft $n + 1$ Knoten (wobei mehrfache Besuche mitgezählt werden). Ein einzelner Knoten kann als Weg der Länge 0 aufgefasst werden.

UNIVERSITÄT BONN

Beispiel

Im oben definierten Graph betrachten wir den Weg $(x, y), (y, z), (z, v)$.



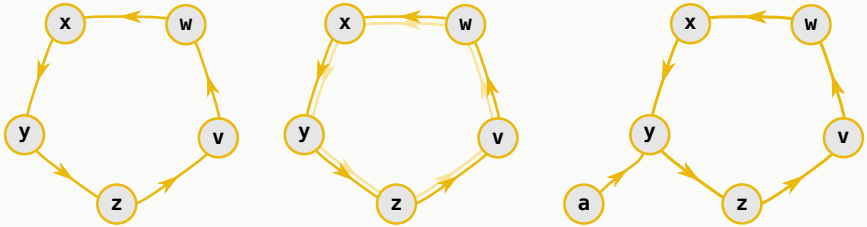
Dieser Weg hat die Länge 3 und besucht die vier Knoten x, y, z und v.

Kreise

In einem gerichteten Graphen definieren wir einen **Kreis** als Weg $e_1, e_2, \ldots, e_n$ der Länge $n \geq 1$ bei dem Start- und Endknoten übereinstimmen.

UNIVERSITÄT BONN

Beispiel



Der linke Weg $(x, y), (y, z), (z, v), (v, w), (w, x)$ ist ein Kreis der Länge 5.

Der mittlere Weg $(x, y), (y, z), (z, v), (v, w), (w, x), (x, y), (y, z), (z, v), (v, w), (w, x)$ ist ein Kreis der Länge 10.

Der rechte Weg $(a, y), (y, z), (z, v), (v, w), (w, x), (x, y)$ ist kein Kreis, da Start- und Endknoten verschieden sind.

UNIVERSITÄT BONN

Gerichte und ungerichtete Graphen

Die eingeführten **gerichteten** Graphen beschreiben **asymmetrische Beziehungen** zwischen Knoten. Beispielsweise kann so die Beziehung *X mag Y* beschrieben werden.

Wir führen nun **ungerichtete** Graphen ein um **symmetrische Beziehungen** zwischen Knoten zu modellieren. Beispielsweise kann so die Beziehung *X und Y sind direkt miteinander verwandt* beschrieben werden.

UNIVERSITÄT BONN

Definition ungerichteter Graph

Sie begegnen in Ihrem Leben vielen Arten von Graphen. In dieser und folgenden Vorlesungen betrachten wir mehrere Arten von **Graphen ohne Mehrfachkanten**.

(Ungerichteter Graph)

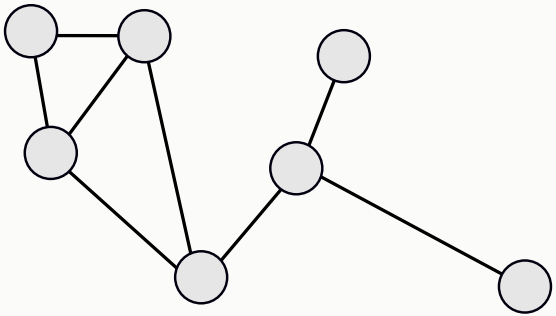
Ein **ungerichteter Graph** $G = (V, E)$ besteht aus einer endlichen Menge von **Knoten** V und einer (endlichen) Menge von **ungerichteten Kanten** $E \subseteq \{\{v, w\} \mid v, w \in V\}$.

Wir sagen dass eine ungerichtete Kante $e = \{v, w\} \in E$ die Knoten v und w verbindet.

UNIVERSITÄT BONN

Beispiel

Dieser Graph hat sieben Knoten und acht Kanten.



Wege und Kreise

Zumindest bildlich ist klar, was Wege, Weglängen und Kreise in einem ungerichteten Graphen sind.

Beim formal sauberen Definieren gibt es allerdings einen Fallstrick. Dieser führt dazu, dass es in der Literatur unterschiedliche Definitionen von Wegen und Kreisen gibt (diese unterscheiden sich nur in für uns uninteressanten Spezialfällen).

In der Vorlesung führen wir eine Definition ein, die formal einfach ist. Allerdings ist das Verknüpfen von zwei Wegen dann nicht einfach definierbar. Für die Interessierten führen wir im Vorlesungsskript eine Definition ein, die formal ein wenig aufwändiger ist. Der Vorteil ist sowohl eine höhrere Flexibilität als auch eine simple Definition der Wegverknüpfung.

UNIVERSITÄT BONN

Wege in ungerichteten Graphen

In einem ungerichteten Graph $G = (V, E)$ ist ein **Weg** definiert als eine alternierende Folge von Knoten und Kanten $x_0, e_1, x_1, e_2, x_2, \ldots, e_n, x_n$ sodass gilt

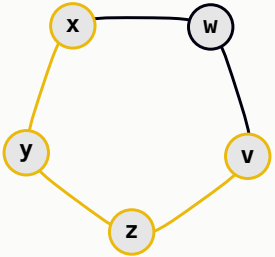
- $x_0, x_1, \ldots, x_n \in V$ und $e_1, e_2, \ldots, e_n \in E$
- $e_i = \{x_{i-1}, x_i\}$ für alle $i = 1, \ldots, n$ und
- jeder Knoten wird höchstens einmal besucht, also $x_i \neq x_j$ für $i \neq j$.

Die **Länge** eines Wegs $x_0, e_1, x_1, e_2, x_2, \ldots, e_n, x_n$ ist n.

UNIVERSITÄT BONN

Beispiel

Wir färben den Weg $x, \{x, y\}, y, \{y, z\}, z, \{z, v\}, v$ ein.



Dieser Weg hat die Länge 3.

UNIVERSITÄT BONN

Kreise in ungerichteten Graphen

In einem ungerichteten Graph $G = (V, E)$ ist ein **Kreis** definiert als eine alternierende Folge von Knoten und Kanten $x_0, e_1, x_1, e_2, x_2, \ldots, x_{n-1}, e_n, x_0$ sodass gilt

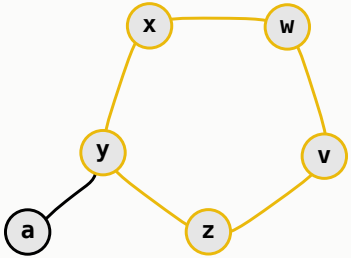
- $x_0, x_1, \ldots, x_{n-1} \in V$ und $e_1, e_2, \ldots, e_n \in E$
- $e_i = \{x_{i-1}, x_i\}$ für alle $i = 1, \ldots, n$ und
- jeder Knoten wird höchstens einmal besucht, also $x_i \neq x_j$ für $i \neq j$.

Die **Länge** eines Kreis $x_0, e_1, x_1, e_2, x_2, \ldots, x_{n-1} e_n, x_0$ ist n.

UNIVERSITÄT BONN

Beispiel

Wir färben den Kreis $x, \{x, y\}, y, \{y, z\}, z, \{z, v\}, v, \{v, w\}, w, \{w, x\}, x$ ein.



Dieser Kreis hat die Länge 5.

Haben Sie Fragen?

Zusammenfassung

Gerichtete Graphen modellieren asymmetrische Beziehungen und ungerichtete Graphen modellieren symmetrische Beziehungen

Wir haben Wege und Kreise in (un)gerichteten Graphen kennen gelernt

Graphen

Erste Graphenalgorithmen: Breiten- und Tiefensuche

Offene Fragen

Was sind erste relevante Graphenalgorithmen?

UNIVERSITÄT BONN

Breitensuche Motivation

Es kommt oft vor, dass die Beziehung zwischen zwei Knoten als direkte Verbindung einer gewissen Länge (z.B. immer Länge 1) interpretiert wird. In diesem Fall wollen wir mit **Breitensuche** Folgendes algorithmisch erreichen.

Wir wollen eine Reihenfolge der von s aus erreichbaren Knoten die erfüllt: Wenn wir die Knoten der Reihenfolge nach betrachten, wird der Abstand zu s entweder größer oder bleibt gleich.

UNIVERSITÄT BONN

Breitensuche Algorithmus

Algorithm: Breitensuche

Eingabe : Ein Graph $G = (V, E)$ und ein Knoten $s \in V$

Ausgabe : Eine Folge $F = v_0, v_1, \ldots \in V$ in oben genannter Reihenfolge

```
1 Erstelle leere Folge F und eine leere Warteschlange (Queue) Q
2 Markiere s als erkundet und füge s in Q und F ein
3 while Q ist nicht leer do
4 |   Sei v = pop(Q)      // Pop entfernt das Element vorn in der Schlange
5 |   for alle Kanten (v, w) do
6 |   |   if w ist noch unerkundet then
7 |   |   |   Markiere w als erkundet und füge w am Ende von Q und F ein
8 return F
```

Die Laufzeit liegt in $\mathcal{O}(|V| + |E|)$ und der Speicherverbrauch in $\mathcal{O}(|V|)$.

Livedemo

UNIVERSITÄT BONN

Tiefensuche Motivation

Ein verwandter Algorithmus ist die **Tiefensuche**. Sie wird zum Beispiel genutzt, um den Ausgang eines Labyrinths intuituv zu bestimmen. Wir wollen mit **Tiefensuche** Folgendes algorithmisch erreichen.

Gegeben ein Startknoten s. Dann sollen die von s aus erreichbaren Knoten als Folge $F = v_0, v_1, \ldots$ notiert werden, so dass gilt: Für jede Kante (v_i, v_j) soll $i < j$ gelten[1].

Dadurch erhalten wir eine **Relation** der (von s aus erreichbaren) Knoten.

[1] Diese Ordnung können wir nur für kreisfreie Graphen erhalten

Tiefensuche Algorithmus

Algorithm: Tiefensuche

Eingabe : Ein Graph $G = (V, E)$ und ein Knoten $s \in V$

Ausgabe : Eine Folge von Knoten $F = v_0, v_1, \ldots$ mit oben genannter Relation

```
1 Erstelle die Folge F = [s]
2 Markiere s als erkundet
3 s for alle Kanten (s, w) mit w unerkundet do
4 |   F' = Tiefensuche(w)    // Wobei die Markierungen mit genannt werden
5 |   Füge F' hinter F ein
6 return F
```

Die Laufzeit liegt in $\mathcal{O}(|V| + |E|)$ und der Speicherverbrauch in $\mathcal{O}(|V|)$.

Livedemo

Verwendung der beiden Suchen

Es gibt viele Varianten der Breiten- und Tiefensuche. Sie haben hier jeweils eine Variante kennen gelernt.

Neben verschiedenen Varianten werden Sie die beiden Suchen auch als Bestandteil von Algorithmen wiedersehen. Dort werden die beiden Suchen oft zur Iteration durch einen Teil der Knoten verwendet.

Haben Sie Fragen?

Zusammenfassung

Objekte und deren Beziehung werden oft mithilfe von Graphen modelliert

Die Beziehungen können dabei durch weitere Eigenschaften beschrieben werden

Sie haben erste Graphen und Graphenalgorithmen kennen gelernt