Anmerkungen zur Übergangsprüfung

Aufgabeneingrenzung

Aufgaben des folgenden Typs werden wegen ihres Schwierigkeitsgrads oder wegen eines ungeeigneten fachlichen Schwerpunkts in der Übergangsprüfung **nicht** gestellt:

2-7; 4-4,2; 4-5; 5-1; 5-3; 6-2; 6-3; 6-5,d; 7-2; 7-4; 8-1; 9-4; 10-2

Alle anderen Aufgaben sollten zur Vorbereitung auf die Übergangsprüfung geübt werden!

Stoffeingrenzung

Eine stoffliche Eingrenzung gibt es nicht:

Die Auswahl der Themen war so, dass sie für das weitere Informatikstudium entweder direkt oder indirekt (wegen der Struktur) relevant sind.

Diskrete Mathematik

Inhalte 11. Vorlesungswoche
Sebastian Iwanowski
FH Wedel

Referenzen zum Nacharbeiten:

Lang 6.2, 6.3, 6.4
Beutelspacher 8.2, 8.3
Meinel 11.2, 11.5
zur Vertiefung: Aigner 6, 7 (7.4: Algorithmus von Dijkstra)
Matousek 3, 4 (3.5: Algorithmus für Eulerwege)

6. Graphentheorie

6.2 Wege in Graphen

Eulerwege

Definition *Eulerweg (Eulerzug)*:

Weg, der alle Kanten im Graphen genau einmal durchläuft

Definition *Eulerkreis*:

Geschlossener Eulerweg (gleiche Anfangs- und Endecke)

Definition *Eulerscher Graph*:

Graph mit einem Eulerkreis

Satz: Ein Graph ist eulersch $\Leftrightarrow$ Jede Ecke hat geraden Grad

6. Graphentheorie

6.2 Wege in Graphen

Eulerwege

Algorithmus zum Auffinden eines Eulerwegs:

- Beginne mit Ecke mit ungeradem Grad (falls nicht vorhanden, dann beliebig). Diese Ecke sei der leere Weg W_0.
- Wiederhole:

  Erweitere W_i zu einem Weg W_{i+1} um eine Kante e_{i+1},
  die an der letzten Ecke von W_i beginnt,
  sodass der Graph $G \setminus W_{i+1}$ zusammenhängend ist
  (isolierte Ecken dürfen vorkommen)

  bis das nicht mehr möglich ist.

Satz: Wenn der vom Algorithmus erzeugte Weg W_k alle Kanten von G enthält, dann ist W_k ein Eulerweg.
Anderenfalls ist kein Eulerweg möglich.

Frage: Wie prüft man nach, ob ein Graph zusammenhängend ist ?

6. Graphentheorie

6.2 Wege in Graphen

<u>Hamiltonwege</u>

Definition *Hamiltonweg*:

Weg, der alle Ecken im Graphen genau einmal durchläuft

Definition *Hamiltonkreis*:

Geschlossener Hamiltonweg (gleiche Anfangs- und Endecke)

Definition *Hamiltonscher Graph*:

Graph mit einem Hamiltonkreis

Problem: Es ist kein effizienter Algorithmus zum Auffinden eines Hamiltonkreises bekannt. ☹

6. Graphentheorie

6.2 Wege in Graphen

Gewichtete Graphen

Definition *Gewichteter (bewerteter) Graph*:

Graph, dessen Kanten mit Gewichten bewertet sind (den Kantenlängen)

Anmerkung: Gewichtete Graphen können auch gerichtet sein.

Darstellung eines gewichteten Graphen im Computer:

- **Adjazenzmatrix:**
  An Position (i,j) steht die Kantenlänge, wenn die Ecken i und j durch eine Kante verbunden sind, sonst 0.

- **Adjazenzliste:**
  In Zeile i stehen die Paare (Eckennummer, Kantenlänge) aller Ecken, die durch eine Kante mit Ecke i verbunden sind.

6. Graphentheorie

6.2 Wege in Graphen

Algorithmus von Dijkstra (Kürzeste Wegeberechnung)

Voraussetzung: Alle Kantenlängen müssen nichtnegativ sein.

Ziel: Es soll der Weg mit der minimalen Kantenlänge von A nach B gefunden werden.

Algorithmus:

- In der Menge **`Berechnet`** sei nur die Ecke A. Markiere A mit Weglänge 0.
  In der Menge **`Unberechnet`** sind alle anderen Ecken des Graphen.
  Markiere die Nachbarn von A mit der Länge der Kante von A nach B
  und alle anderen Ecken mit Weglänge ∞.
- Wiederhole:
  Wähle die Ecke V aus **`Unberechnet`** mit der kleinsten Markierung
  und verschiebe sie in die Menge **`Berechnet`**.
  Betrachte alle Nachbarn N von V aus **`Unberechnet`**:
  Ersetze die Markierung von N durch das Minimum seiner bisherigen Markierung
  und der Summe der Markierung von V plus der Länge der Kante von V zu N.

  bis V = B

6. Graphentheorie

6.2 Wege in Graphen

Algorithmus von Dijkstra (Kürzeste Wegeberechnung)

Satz: Die Markierungen aller Ecken V der Menge **`Berechnet`** entsprechen der Länge des kürzesten Wegs von A nach V.

Erweiterung zur *Ausgabe* des kürzesten Wegs:

- Wiederhole:

  Wähle die Ecke V aus **`Unberechnet`** mit der kleinsten Markierung und verschiebe sie in die Menge **`Berechnet`**.

  Betrachte alle Nachbarn N von V aus **`Unberechnet`**:

  Ersetze die Markierung von N durch das Minimum seiner bisherigen Markierung und der Summe der Markierung von V plus der Länge der Kante von V zu N.
  Falls die Markierung aktualisiert werden muss, mache V zum **Vorgänger** von N.

  bis V = B

- Sammle nacheinander alle Vorgänger von B bis A auf
  und gib den Weg in umgekehrter Reihenfolge wieder aus.

Satz: Der Algorithmus von Dijkstra berechnet nicht nur den kürzesten Weg von A nach B, sondern auch alle kürzesten Wege von A zu allen anderen Ecken, die näher an A sind als B.

6. Graphentheorie

6.3 Bäume

Definition *Baum*:

Ein Baum ist ein zusammenhängender Graph ohne Kreise.

Definition *Wald*:

Ein Wald ist ein Graph ohne Kreise (nicht notwendig zusammenhängend)

Satz: Folgende Aussagen sind äquivalent:

- G ist ein Baum.
- G ist ein Graph ohne Kreise mit maximal vielen Kanten
  (d.h. beim Hinzufügen einer beliebigen Kante entsteht immer ein Kreis).
- G ist ein zusammenhängender Graph mit n-1 Kanten
  (wobei n die Eckenzahl des Graphen ist).

6. Graphentheorie

6.3 Bäume

Definition *Suchbaum*:

- Ein *Suchbaum* ist ein Baum, in dem ein Knoten als Wurzel ausgezeichnet ist.
- Die *Suchtiefe eines Knotens* ist die Länge des Weges zur Wurzel.
- Die *Suchtiefe eines Suchbaums* ist die maximale Suchtiefe seiner Knoten.
- Die Nachbarn eines Knoten, die eine größere Suchtiefe haben als dieser, heißen *Kinder* dieses Knoten.
- Der (eindeutige) Nachbarn eines Knoten, der eine geringere Suchtiefe hat als dieser, heißt *Elternteil* dieses Knoten.
- Ein *Blatt* ist ein Knoten mit Grad 1 (aber niemals die Wurzel!), also ein Knoten ohne Kinder.

Suchbäume mit maximaler Kinderzahl:

- Ein binärer *Suchbaum* ist ein Suchbaum, in dem alle Knoten maximal 2 Kinder haben.
- Ein ternärer *Suchbaum* ist ein Suchbaum, in dem alle Knoten maximal 3 Kinder haben.
- Ein d-ärer *Suchbaum* ist ein Suchbaum, in dem alle Knoten maximal d Kinder haben.

Satz: Ein d-ärer Suchbaum mit n Blättern hat mindestens Suchtiefe $\log_d n$

6. Graphentheorie

6.3 Bäume

Definition *Spannender Baum (Gerüst)*:

Ein Spannender Baum (Gerüst) eines Graphen ist ein Teilgraph,
der selbst ein Baum ist und alle Ecken des ursprünglichen Graphen enthält.

Konstruktion eines spannenden Baums für einen beliebigen Graphen G:

- Beginne mit dem leeren Wald W, bestehend aus keiner Kante.
- Wiederhole für alle Kanten e_1, e_2, ..., e_m des Graphen G (Reihenfolge beliebig):
  Untersuche, ob e_i zu W hinzugefügt werden kann,
  sodass W weiterhin kreisfrei bleibt:
  Falls ja, füge e_i zu W hinzu.

  bis W aus n-1 Kanten besteht (n sei die Anzahl der Ecken von G).

Satz: Der so konstruierte Wald W ist ein spannender Baum für G.

6. Graphentheorie

6.3 Bäume

Definition *Minimaler Spannender Baum*:

Ein Minimaler Spannender Baum eines gewichteten Graphen ist ein spannender Baum, dessen Gesamtkantenlänge minimal ist.

Algorithmus von Kruskal:

Konstruktion eines minimalen spannenden Baums für ein beliebiges G:

- Beginne mit dem leeren Wald W, bestehend aus keiner Kante.
- Wiederhole für alle Kanten e_1, e_2, ..., e_m des Graphen G (Reihenfolge sortiert):
  Untersuche, ob e_i zu W hinzugefügt werden kann,
  sodass W weiterhin kreisfrei bleibt:
  Falls ja, füge e_i zu W hinzu.
  bis W aus n-1 Kanten besteht (n sei die Anzahl der Ecken von G).

Satz: Der so konstruierte Wald W ist ein minimaler spannender Baum für G.