



Institut für Theoretische Informatik
Peter Widmayer
Beat Gfeller

Prüfung

Datenstrukturen und Algorithmen

D-INFK

19. August 2009

Name, Vorname: _____

Stud.-Nummer: _____

Ich bestätige mit meiner Unterschrift, dass ich diese Prüfung unter regulären Bedingungen ablegen konnte und dass ich die untenstehenden Hinweise gelesen und verstanden habe.

Unterschrift: _____

Hinweise:

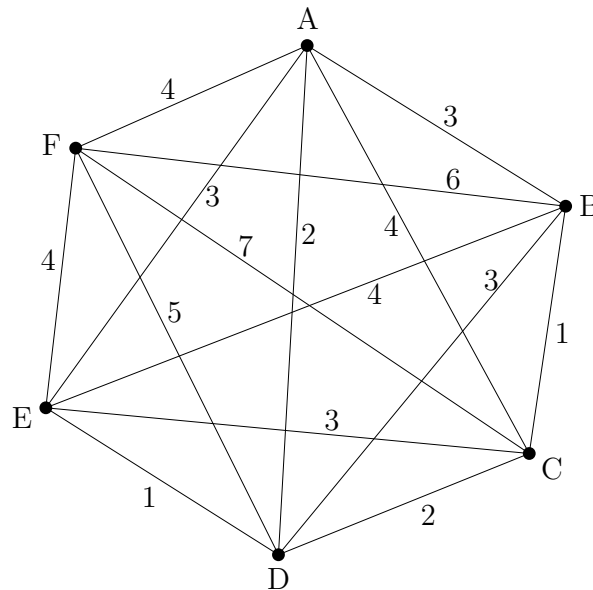
- Ausser einem Wörterbuch dürfen Sie keine Hilfsmittel verwenden.
- Bitte schreiben Sie Ihre StudentInnen-Nummer auf **jedes** Blatt.
- Melden Sie sich bitte **sofort**, wenn Sie sich während der Prüfung in irgendeiner Weise bei der Arbeit gestört fühlen.
- Bitte verwenden Sie für jede Aufgabe ein neues Blatt. Pro Aufgabe kann nur eine Lösung angegeben werden. Ungültige Lösungsversuche müssen klar durchgestrichen werden.
- Bitte schreiben Sie **lesbar** mit blauer oder schwarzer Tinte. Wir werden nur bewerten, was wir lesen können.
- Die Prüfung dauert 120 Minuten. **Keine Angst!** Wir rechnen nicht damit, dass irgendjemand alles löst! Sie brauchen bei weitem nicht alle Punkte, um die Bestnote zu erreichen.

Viel Erfolg!

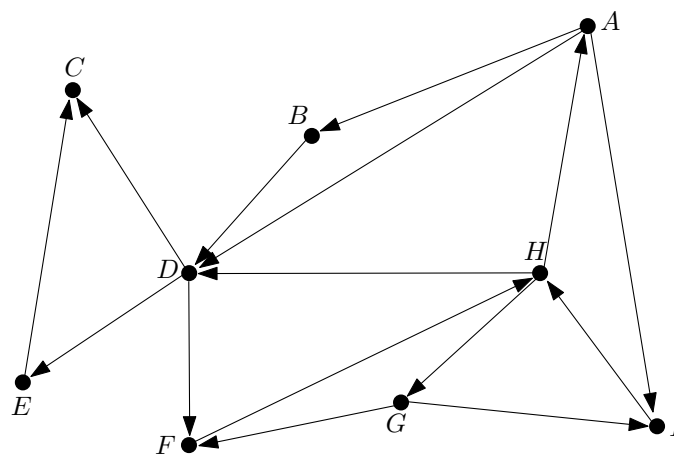
Stud.-Nummer: _____

Aufgabe	1	2	3	4	5	Σ
Mögl. Punkte	9	7	9	9	9	43
Punkte						

- 1 P d) Berechnen Sie im untenstehenden Graphen *mittels des Algorithmus von Christofides* eine Rundreise, deren Länge höchstens das $\frac{3}{2}$ -fache der Länge der Optimalen Tour ist. Sie müssen diese Schranke nicht beweisen. Schreiben Sie die Reihenfolge auf, in der die Knoten in dieser Rundreise besucht werden.



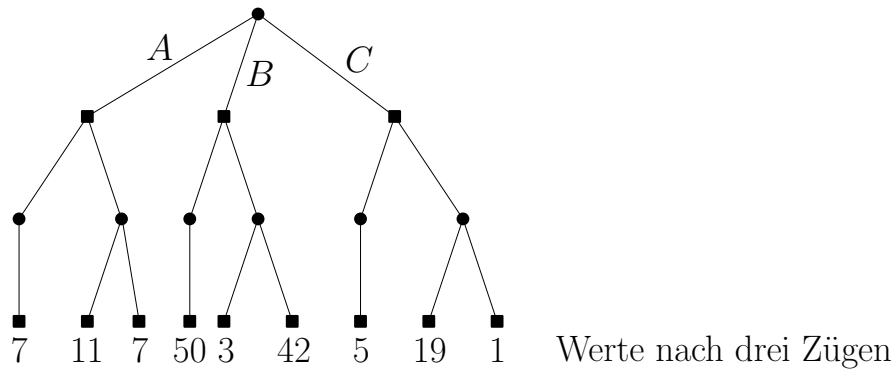
- 1 P e) Der folgende gerichtete Graph wird mit Tiefensuche traversiert. Die Suche startet beim Knoten A. Geben Sie eine Reihenfolge an, in der die Knoten erreicht werden können.



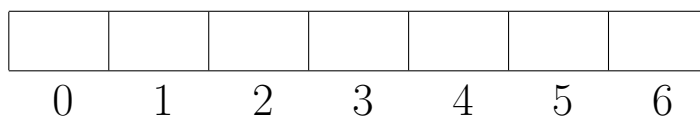
- 1 P f) Gegeben sind acht Schlüssel mit der jeweiligen Anzahl der Zugriffe. Erstellen Sie mit Hilfe des Huffman-Algorithmus einen optimalen Codierungsbaum, und zeichnen Sie diesen.

Schlüssel	a	b	c	d	e	f	g	h
Anzahl Zugriffe	22	1	5	3	3	2	1	9

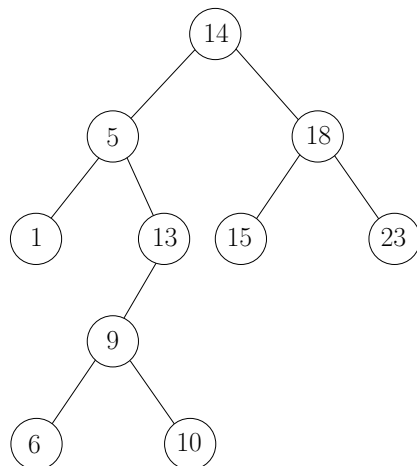
- 1 P g) Der folgende Baum stellt eine Spielbaumsuche der Tiefe drei dar, in der jeder mögliche resultierende Zustand mit einem Wert bewertet wird. Ein höherer Wert bedeutet, dass der Zustand besser für den aktuellen Spieler ist (und schlechter für den zweiten Spieler). Welcher nächste Zug (*A*, *B* oder *C*) ist der beste für den aktuellen Spieler?



- 1 P h) Fügen Sie die Schlüssel 1, 12, 4, 17, 89, 11, 5 in dieser Reihenfolge mittels Offenem Hashing in die Hashtabelle ein, und benutzen Sie dabei Quadratisches Sondieren. Die zu verwendende Hash-Funktion ist $h(k) = (k \bmod 7)$.



- 1 P i) Zeichnen Sie den Splay-Tree, der aus dem unten gezeigten entsteht, nachdem man auf den Knoten 9 zugegriffen hat (und die zugehörigen Splay-Operationen ausgeführt wurden).



Aufgabe 2:

- 1 P** a) Geben Sie für die untenstehenden Funktionen eine **Reihenfolge** an, so dass folgendes gilt: Wenn Funktion f links von Funktion g steht, so gilt $f \in O(g)$.

Beispiel: Die drei Funktionen n^3, n^7, n^9 sind bereits in der entsprechenden Reihenfolge, da $n^3 \in O(n^7)$ und $n^7 \in O(n^9)$ gilt.

- $\log\left(\frac{n^3}{2}\right)$
- $2^{\sqrt{n}}$
- $\sum_{i=1}^n i$
- $\log^2 n$
- \sqrt{n}
- $\frac{n}{\log n}$

- 3 P** b) Gegeben ist die folgende Rekursionsgleichung:

$$T(n) := \begin{cases} 2T\left(\frac{n}{4}\right) + 3n + 5 & n > 1 \\ 2 & n = 1 \end{cases}$$

Geben Sie eine geschlossene (d.h. nicht-rekursive) und möglichst einfache Formel für $T(n)$ an und beweisen Sie diese mit vollständiger Induktion.

Hinweise:

- (1) Sie können annehmen, dass n eine Potenz von 4 ist.
- (2) Für $q \neq 1$ gilt: $\sum_{i=0}^k q^i = \frac{q^{k+1}-1}{q-1}$.

- 1 P** c) Geben Sie die asymptotische Laufzeit in Abhängigkeit von $n \in \mathbb{N}$ für folgenden Algorithmus in Theta-Notation an (Sie müssen Ihre Antwort nicht begründen):

```
from i := n until i < 1 loop
  from j := 1 until j > i loop
    x := i + 1
    j := j + 1
  end
  i := i - 1
end
```

- 1 P** d) Geben Sie die asymptotische Laufzeit in Abhängigkeit von $n \in \mathbb{N}$ für folgenden Algorithmus in Theta-Notation an (Sie müssen Ihre Antwort nicht begründen):

```
from
  i := n
  s := 1
  k := 1
until i < 1 or i > 2*n loop
  i := i + s*k
  s := -s
  k := 2*k
end
```

- 1 P** e) Geben Sie die asymptotische Laufzeit in Abhängigkeit von $n \in \mathbb{N}$ für folgenden Algorithmus in Theta-Notation an (Sie müssen Ihre Antwort nicht begründen):

```
from
  i := 1
  k := 1
until i > n loop
  from j := 1 until j > k loop
    j := j + 1
  end
  if k > n then
    k := k // 2
  else
    k := k * 2
  end
  i := i + 1
end
```

Aufgabe 3:

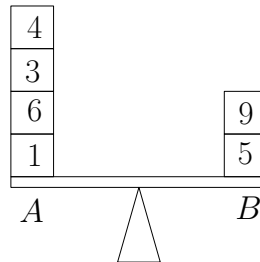
Ein Hotel benötigt ein System, um die Buchungen für ihre Präsidenten-Suite abzuwickeln. Diese Suite kann für eine beliebige Anzahl Nächte gemietet werden. Natürlich kann die Suite aber für jede Nacht nur an einen Kunden vermietet werden. Die Suite kann nahtlos vermietet werden, d.h. wenn am Tag y ein Kunde abreist, kann sie noch am gleichen Tag ein anderer Kunde beziehen. Das System muss drei Operationen unterstützen: Buchungen eintragen (wenn eine Reservation getätigt wird), löschen (wenn eine Reservation annulliert wird), und prüfen ob die Suite für ein gegebenes Paar von Ankestag und Abreisetag noch verfügbar ist. Eine *Buchung* (x, y) besteht aus einem Ankestag x und einem Abreisetag y , wobei x und y natürliche Zahlen sind, und $x < y$.

- 3 P** a) Als ersten Planungsschritt sollen Sie eine Datenstruktur angeben, in der n bereits getätigte Buchungen gespeichert werden können, so dass die Anfrage, ob eine gewünschte Buchung (x', y') noch möglich ist, effizient beantwortet werden kann. Beschreiben Sie kurz in Worten
- wie Ihre Datenstruktur konstruiert ist,
 - wie sie für n bestehende Buchungen aufgebaut wird, und welche Laufzeit dieser Aufbau benötigt,
 - und wie die Abfrage für eine beliebige gewünschte Buchung (x', y') effizient beantwortet werden kann. Geben Sie auch hier die Laufzeit für eine beliebige Abfrage an.
- 6 P** b) Nun soll das System verallgemeinert werden, so dass die Menge der Buchungen nicht mehr statisch ist. Das heisst, man will eine Datenstruktur, welche folgende Operationen auf der (anfängs leeren) Menge der Buchungen effizient unterstützt:
- **Insert** (x, y) : eine Buchung (x, y) einfügen
 - **Delete** (x, y) : eine Buchung (x, y) entfernen
 - **Query** (x, y) : feststellen, ob eine Buchung (x, y) für die aktuelle Menge von Buchungen noch möglich ist

Alle Operationen sollen möglichst effizient in Abhängigkeit der Anzahl Buchungen in der aktuellen Menge sein. Beschreiben Sie Ihre Datenstruktur in Worten, und die Implementation aller drei Operationen in Pseudocode. Ihr Pseudocode muss in Java- Eiffel- oder C++-ähnlicher Notation geschrieben sein. Geben Sie zudem die Laufzeiten der Operationen **Insert**, **Delete** und **Query** an.

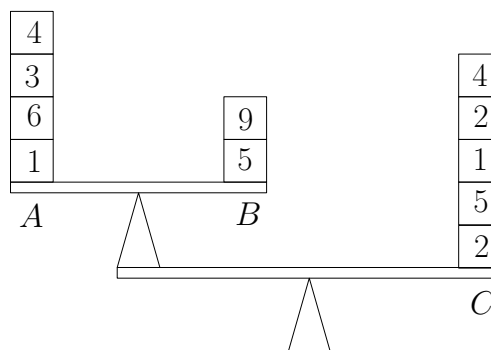
Aufgabe 4:

Eine Künstlerin plant eine Skulptur zum Thema “Gleichgewicht” (siehe die Abbildung unten). Sie hat n Würfel zur Verfügung, welche beliebig hoch aufeinander gestapelt werden können. Die Würfel sind alle gleich gross, haben aber verschiedene Gewichte: Jeder Würfel i hat ein ganzzahliges Gewicht w_i . Die vorhandenen n Würfel sollen auf die zwei Seiten (A und B) einer Waage verteilt werden, so dass diese im Gleichgewicht ist. Dies ist genau dann der Fall, wenn die Summe der Gewichte auf der linken Seite gleich der Summe auf der rechten Seite ist. Zu beachten ist, dass *alle* n Würfel benutzt werden sollen.



- 5 P** a) Entwerfen Sie einen Algorithmus nach dem Muster der dynamischen Programmierung, der feststellt, ob für eine gegebene Menge von n Würfeln mit Gewichten w_1, w_2, \dots, w_n ein Gleichgewicht möglich ist. Schreiben Sie die Rekursionsgleichung des entsprechenden dynamischen Programms auf, und geben Sie zudem die Laufzeit Ihres Algorithmus an.
- 1 P** b) Beschreiben Sie kurz in Worten, wie der Algorithmus in a) erweitert werden kann, um eine Aufteilung der Würfel auszugeben, welche ein Gleichgewicht erreicht, falls eine solche existiert.
- 3 P** c) Da die erste Skulptur ein Erfolg war, plant die Künstlerin eine weitere (siehe die Abbildung unten). Für diese müssen die Würfel in drei Mengen A , B und C aufgeteilt werden, so dass folgende Bedingungen gelten:
- Das Gesamtgewicht der Menge A ist gleich dem Gesamtgewicht der Menge B .
 - Das Gesamtgewicht der Menge C ist gleich dem Gesamtgewicht der Menge A plus dem Gesamtgewicht der Menge B .

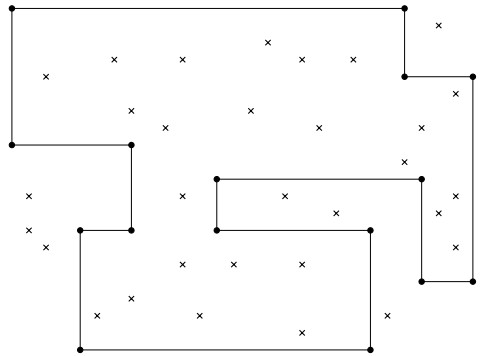
Wiederum müssen alle Würfel verwendet werden. Entwerfen Sie ein dynamisches Programm, mit dem festgestellt werden kann, ob eine solche Aufteilung möglich ist, und geben Sie die Laufzeit Ihres Algorithmus an.



Aufgabe 5:

Ein Schweizer Bauer besitzt eine stattliche Anzahl von Kühen. Diese sind alle innerhalb eines einzigen Geheges mit geradlinigen Zäunen untergebracht. Um zu prüfen, ob noch alle Kühe im Gehege sind, will der Bauer die Kühe im Gehege zählen. Wir nehmen an, dass der Bauer die Positionen all seiner Kühe bestimmen kann, zum Beispiel mit Hilfe eines GPS-Empfängers für jede Kuh. Natürlich kennt der Bauer auch die genauen Positionen der Ecken seines Zauns. Zunächst nehmen wir an, dass jede Ecke des Zauns einen rechten Winkel bildet, und somit alle Kanten orthogonal zueinander sind.

In dieser Aufgabe soll ein Algorithmus entwickelt werden, um die (momentane) Anzahl der Kühe innerhalb des Geheges zu zählen. Genauer: Gegeben ist ein einfaches, geschlossenes Polygon mit n *orthogonalen* Kanten (das Gehege), sowie die Position von m Punkten in der Ebene (die Kühe). Die Kühe sind anhand ihrer x - und y -Koordinate gegeben. Das Polygon ist anhand der (ungeordneten) Liste aller Kanten gegeben, wobei jede Kante als geordnetes Paar von Ecken (a, b) gegeben ist, wobei das Polygoninnere rechts liegt, wenn man von a aus in Richtung b schaut. Gesucht ist die Anzahl der Punkte, welche innerhalb des Polygons liegen. Der Einfachheit halber nehmen wir an, dass kein Punkt genau auf dem Rand des Polygons liegt. Eine Beispiel-Instanz ist auf der untenstehenden Abbildung dargestellt.



- 5 P** a) Entwerfen Sie einen möglichst effizienten Algorithmus, der für ein gegebenes orthogonales Polygon mit n Kanten und eine gegebene Menge von m Punkten die Anzahl Punkte im Polygon berechnet. Beschreiben Sie Ihren Algorithmus in Worten und/oder in Pseudocode.
- 1 P** b) Welche Laufzeit hat Ihr Verfahren im schlechtesten Fall, in Abhängigkeit von m und n ?
- 3 P** c) Nun soll das Verfahren erweitert werden, so dass die Kanten des Polygons nicht mehr orthogonal sein müssen. Beschreiben Sie in Worten, welche Teile Ihrer Lösung in welcher Weise erweitert werden müssen. Welche Laufzeit hat die erweiterte Lösung?

