Berlin, 21. Februar 2013

Name: .... **Matr.-Nr.: ....**

Klausur Informatik-Propädeutikum

(Niedermeier/Hartung/Nichterlein, Wintersemester 2012/13)

1	
2	
3	
4	
5	
6	
7	
8	
9	
Σ	

Bearbeitungszeit:	90 min.
max. Punktezahl:	60 Punkte
min. Punktezahl zum Bestehen:	20 Punkte

Allgemeine Hinweise:

- Es sind keinerlei Hilfsmittel erlaubt.
- Benutzen Sie keinen Bleistift, sondern einen Kugelschreiber oder Füller in der Farbe schwarz oder blau.
- Beschriften Sie jedes Blatt mit Vor- und Nachnamen sowie Matrikelnummer.
- **Falls in der Aufgabenstellung nicht explizit ausgeschlossen, sind alle Antworten zu begründen! Antworten ohne Begründung erhalten 0 Punkte.**

Viel Erfolg!

Name: .... **Matr.-Nr.: ....**

Aufgabe 1: **Stable Matching (Gale und Shapley)** (2+2+2+2 Punkte)

Sei die Menge der Männer $M = \{A, B, C\}$ und die Menge der Frauen $F = \{1, 2, 3\}$ mit folgenden Präferenzlisten gegeben:

$$\begin{array}{ll} A: 1 > 2 > 3, & 1: B > C > A \\ B: 2 > 3 > 1, & 2: C > A > B \\ C: 1 > 2 > 3, & 3: A > B > C \end{array}$$

(a) Geben Sie eine „männeroptimale“ stabile Zuordnung an (ohne Begründung).

——————Lösung——————

Lösung mittels G&C-Algorithmus:

1. (A,), (B,), (C,)
2. (A,1),(B,), (C,)
3. (A,1),(B,2), (C,)
4. (A,),(B,2), (C,1)
5. (A,2),(B,), (C,1)
6. (A,2),(B,3), (C,1)

————————————————————

(b) Geben Sie eine „frauenoptimale“ stabile Zuordnung an (ohne Begründung).

——————Lösung——————

1. (1,B), (2,), (3,)
2. (1,B), (2,C), (3,)
3. (1,B), (2,C), (3,A)

————————————————————

(c) Gibt es immer eine stabile Zuordnung, welche sowohl männeroptimal als auch frauenoptimal ist? Hinweis: Sie können z. B. beweisen, dass dies nicht stimmt indem Sie Präferenzlisten angeben für welche dies nicht der Fall sein kann.

——————Lösung——————

Nein!
Es gibt nur eine stabile Zuordnung, welche männeroptimal ist und auch nur eine welche frauenoptimal ist. Obige Präferenzlisten haben zwei unterschiedlichen stabilen Zuordnungen, die jeweils männeroptimal und frauenoptimal sind. Das heißt, wir haben ein Gegenbeispiel das keine stabile Zuordnung, die sowohl männeroptimal als auch frauenoptimal ist.

————————————————————

(d) Geben Sie Präferenzlisten mit mindestens jeweils 3 Männern und Frauen an, welche eine stabile Zuordnung zulassen, welche sowohl männer- als auch frauenoptimal ist.

——————Lösung——————

$$
\begin{aligned}
A &: 1 > 2 > 3, \qquad & 1 &: A > B > C \\
B &: 2 > 3 > 1, \qquad & 2 &: B > C > A \\
C &: 3 > 1 > 2, \qquad & 3 &: C > A > B
\end{aligned}
$$

Offensichtlich ist $(A, 1), (B, 2), (C, 3)$ stabil, männeroptimal und auch frauenoptimal, da jeder seinen Wunschpartner kriegt.

Hinweis: Eine stabile Zuordnung wird **männer-(frauen-)optimal** genannt, wenn es keine andere stabile Zuordnung gibt, welche mindestens einen Mann (Frau) besser stellt, d. h. er (sie) bekommt eine(n) Frau (Mann) mit höherer Präferenz. Der Gale-Shapley Algorithmus liefert immer eine männeroptimale Lösung.

Zur Erinnerung:

Eine Zuordnung der Männer M zu den Frauen W ist eine **stabile** Zuordnung, falls für jeden $m \in M$ und jede $w \in W$ die nicht m zugeordnet ist gilt:

(a) m zieht die ihm zugeordnete w' gegenüber w vor, oder

(b) w zieht den ihr zugeordneten m' gegenüber m vor.

Propose&Reject [Gale, Shapley 1962]

Initialisiere alle $m \in M$ und alle $w \in W$ als „frei“
while $\exists m \in M :$ m ist frei und $\exists w \in W$ der m noch keinen Antrag gemacht hat
    $w \leftarrow$ erste noch „unbeantragte“ Frau in m’s Präferenzfolge
    if w ist frei **then:**
        (m, w) wird Paar, $m \leftarrow$„verlobt“, $w \leftarrow$„verlobt“
    else if w zieht m ihrem aktuellen „Verlobten“ m' vor **then:**
        (m, w) wird Paar, $m \leftarrow$„verlobt“, $w \leftarrow$„verlobt“, $m' \leftarrow$„frei“
    else: w lehnt m ab

Name: .... **Matr.-Nr.: ....**

Aufgabe 2: **Heuristiken für Vertex Cover** (2 + 2 + 3 Punkte)

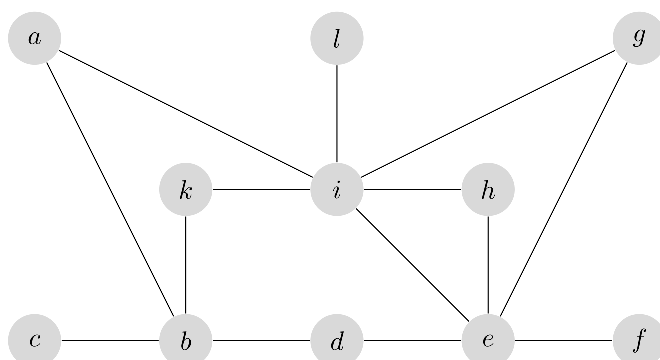
Vertex Cover wurde in der Vorlesung wie folgt definiert:

Vertex Cover
Eingabe: Ein ungerichteter Graph G.
Aufgabe: Finde eine kleinstmögliche Knotenmenge in G, sodass jede Kante mindestens einen dieser Knoten als Endpunkt hat.

Gegeben sind die folgenden beiden Heuristiken für Vertex Cover und folgender Graph G:

Heuristik 1: Solange es eine Kante gibt, wähle irgendeine, nehme beide Endpunkte dieser Kante in die Lösungsmenge und lösche diese zwei Knoten aus dem Graph.

Heuristik 2: Solange es eine Kante gibt, nehme einen Knoten mit der höchsten Anzahl anliegender Kanten in die Lösungsmenge und lösche ihn aus dem Graph.



(a) Geben Sie für den gegebenen Graph (siehe obiges Bild) jeweils eine von Heuristik 1 und eine von Heuristik 2 erzeugte Lösungsmenge an (ohne Begründung).

——————Lösung——————

Heuristik 1: z.B. $\{c, b, d, e, k, i\}$ (nicht eindeutig)
Heuristik 2: $\{b, i, e\}$

(b) Ist eine der zwei erzeugten Lösungsmengen optimal (d.h. kleinstmöglich)?

——————Lösung——————

Die Lösung von Heuristik 2 ist optimal: Die Knoten c, f, l haben alle jeweils genau einen Nachbarn und diese Nachbarn sind verschieden $\rightsquigarrow$ mindestens 3 Knoten in einer Lösung benötigt.

(c) Nur eine der beiden Heuristiken ist eine Faktor-2-Approximation, d.h. die von dieser Heuristik gelieferten Lösungsmengen sind bei beliebigen Eingabegraphen maximal doppelt so groß wie die optimalen Lösungsmengen. Welche der beiden Heuristiken ist die Faktor-2-Approximation?

——————Lösung——————

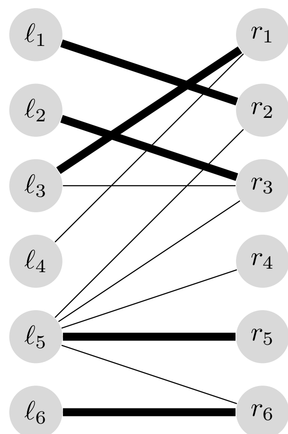
Nur Heuristik 1 ist eine Faktor-2-Approximation: für jede der betrachteten Kanten muss mindestens einer der beiden anliegenden Knoten in einer Lösung enthalten sein - die Heuristik nimmt beide $\rightsquigarrow$ Faktor-2-Approximation.

Name: .... **Matr.-Nr.: ....**

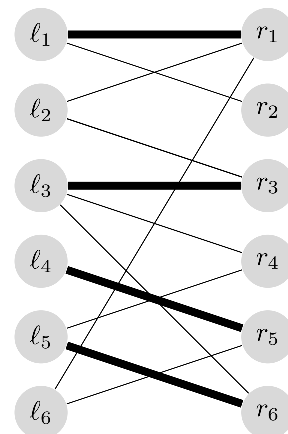
Aufgabe 3: **Bipartites Matching** (6 Punkte)

Gegeben sind folgende zwei bipartite Graphen G_1 und G_2.

G_1



G_2

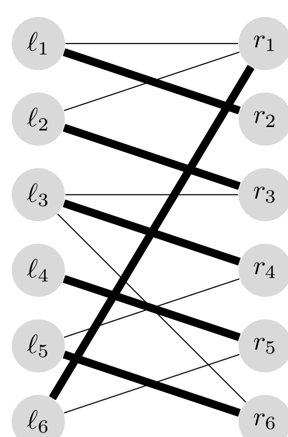


Die fett gezeichneten Kanten geben ein Matching in den jeweiligen Graphen an. Sind die beiden angegeben Matchings größtmöglich? Wenn nicht, geben Sie ein größtmögliches Matching an.

——————Lösung——————

Das Matching in G_1 ist größtmöglich: Da der Graph 12 Knoten hat, kann ein größtmögliches Matching aus maximal 6 Kanten bestehen. Da r_4 und r_5 beide nur den einen Nachbarn ℓ_5 haben, kann es kein Matching der Größe 6 geben. Also ist das Matching mit 5 Kanten größtmöglich.

Das Matching in G_2 ist nicht größtmöglich: Folgendes Matching ist größer.



Name: .... **Matr.-Nr.: ....**

Aufgabe 4: **Algorithmische Komplexität** (4+4 Punkte)

Für aussagenlogische Variablen spricht man von einer Klausel, wenn eine beliebige Anzahl von Literalen (für eine Variable x sind x und $\overline{x}$ die dazugehörigen Literale) durch ein logisches ODER verknüpft wird. Eine aussagenlogische Formel befindet sich in konjunktiver Normalform, falls sie aus durch ein logisches UND verketteten Klauseln besteht.

Beispiel: $\underbrace{(x_1 \vee \overline{x_2} \vee x_3 \vee x_5)}_{\text{Klausel}} \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_3} \vee \overline{x_4})$

Eine aussagenlogische Formel heißt **erfüllbar**, falls es eine Belegung der Variablen mit 0 oder 1 gibt, sodass die Formel sich zu 1 auswertet. Obige Formel ist erfüllbar z. B. durch die Belegung $x_1 = 1, \overline{x_3} = 1$ (x_4 und x_5 beliebig).

Betrachten Sie die beiden folgenden Probleme.

SAT
Eingabe: Aussagenlogische Formel F in konjunktiver Normalform.
Frage: Ist F erfüllbar?

3-SAT
Eingabe: Aussagenlogische Formel F in konjunktiver Normalform mit höchstens drei Literalen pro Klausel.
Frage: Ist F erfüllbar?

Beachten Sie, dass beim SAT-Problem jede Klausel beliebig viele Variablen beinhalten kann.

Wir nehmen nun an wir hätten einen Algorithmus $\mathcal{A}^{\text{3-SAT}}$, welcher das 3-SAT-Problem in polynomieller Zeit löst. Wir wollen nun auch das SAT-Problem für eine Formel F in polynomieller Zeit mithilfe des folgenden Algorithmus lösen:

1. Konstruiere eine Formel F', indem jede Klausel $(\ell_1 \vee \ell_2 \vee \ldots \vee \ell_m)$ mit $m \geq 4$ in F ersetzt wird durch
$$(\ell_1 \vee \ell_2 \vee z_1) \wedge (\overline{z_1} \vee \ell_3 \vee z_2) \wedge \ldots \wedge (\overline{z_{m-3}} \vee \ell_{m-1} \vee \ell_m),$$
wobei $z_1, z_2, \ldots, z_{m-3}$ neue Variablen sind, welche bislang noch nicht in F enthalten sind (man führt insbesondere für jede zu ersetzende Klausel neue z-Variablen ein).
2. Löse das 3-SAT-Problem für F' mithilfe des Algorithmus $\mathcal{A}^{\text{3-SAT}}$ und gebe das Ergebnis aus.

Beweisen Sie, dass obiger Algorithmus korrekt ist. Gehen Sie dazu wie folgt vor: Sei X die Menge der Variablen in Formel F und Z die Menge der neu hinzugefügten Variablen in Formel F'.
1) Beweisen Sie, dass wenn F erfüllbar ist, dann auch F'.

————————Lösung————————

Angenommen, F ist erfüllbar und $\beta : X \to \{0,1\}$ ist eine Variablenbelegung, die F zu 1 auswertet. Wir beweisen, dass die Variablenbelegung $\beta' : X \cup Z \to \{0,1\}$ die Formel F' zu 1 auswertet, wobei

- $\beta'(x) = \beta(x)$ falls $x \in X$, und

- für die zur Klausel $(\ell_1 \vee \ell_2 \vee \ldots \vee \ell_m)$ mit $m \geq 4$ in F zugehörigen neuen Variablen $z_1, z_2, \ldots, z_{m-3}$:

  Sei l_i das Literal mit dem kleinsten Index, sodass l_i zu 1 auswertet (So ein Literal muss existieren, denn β ist eine erfüllbare Variablenbelegung für Formel F).

  Dann setze die neuen Variablen wie folgt:

  (a) $\beta'(z_j) = 0, \forall 1 \leq j \leq m-3$, falls $i \leq 2$;
  (b) $\beta'(z_j) = 1, \forall 1 \leq j \leq m-3$, falls $i \geq m-1$;
  (c) $\beta'(z_j) = 1, \forall 1 \leq j \leq i-2$ und $\beta'(z_j) = 0, \forall i-1 \leq j' \leq m-3$, sonst.

  Für die Korrektheit betrachten wir noch einmal die dazugehörige Formel $F'' = (\ell_1 \vee \ell_2 \vee z_1) \wedge (\overline{z_1} \vee \ell_3 \vee z_2) \wedge \ldots \wedge (\overline{z_{m-3}} \vee \ell_{m-1} \vee \ell_m)$.

  Für den Fall, dass $i \leq 2$: Die erste Klausel in F'' ist erfüllt durch β; Jede neue Variable z_j kommt genau einmal negativ vor in allen restlichen Klauseln, also sind alle Klauseln durch β' erfüllt.

  Analog gilt das für den Fall, dass $i \geq m-1$.

  Ansonsten kommt jede Variable z_j mit $j \leq i-2$ genau einmal positiv vor in den ersten $i-2$ Klauseln; jede Variable z_j mit $j \geq i-1$ kommt genau einmal negativ vor in den letzten $m-3-(i-1)$ Klauseln; die $i-1$-ste Klausel ist erfüllt durch β.

  Offsichtlich sind alle Klauseln mit weniger als 4 Literalen in F die auch in F' sind durch β' zu 1 ausgewertet.

---

2) Beweisen Sie, dass wenn F' erfüllbar ist, dann auch F.

——————Lösung——————

Sei $\beta' : X \cup Z \to \{0, 1\}$ eine Variablenbelegung, die Formel F' zu 1 auswertet. Dann ist die Variablenbelegung β', die auf die Variablenmenge X eingeschränkt ist, eine erfüllbare Belegung für Formel F. Beweis per Widerspruch: Angenommen, eine Klausel $\ell_1 \vee \ell_2 \vee \ldots \vee \ell_m$ mit $m \geq 4$ ist nicht durch β erfüllt. (Alle Klauseln mit weniger als 4 Literalen sind offensichtlich erfüllt.) Um aber die erste Klausel in $F'' = (\ell_1 \vee \ell_2 \vee z_1) \wedge (\overline{z_1} \vee \ell_3 \vee z_2) \wedge \ldots \wedge (\overline{z_{m-3}} \vee \ell_{m-1} \vee \ell_m)$ zu erfüllen, muss $\beta'(z_1) = 1$ sein. Daraus folgt, dass $\beta'(z_2) = 1$, usw. Aber die letzte Klausel in F'' ist dann nicht mehr erfüllt. Widerspruch!

Name: .... **Matr.-Nr.: ....**

Aufgabe 5: **Ackermannfunktion** (5 Punkte)

Die Ackermann-Funktion f ist wie folgt rekursiv definiert:

$$\begin{aligned} f(k,1) &= 2, \\ f(1,n) &= n+2 \text{ für } n>1, \\ f(k,n) &= f(k-1, f(k,n-1)) \text{ für } k,n>1. \end{aligned}$$

Berechnen Sie $f(3,3)$.

———————Lösung———————

$f(3,3) = f(2, f(3,2)) = f(2,4) = f(1, f(2,3)) = f(1,6) = 8.$

$f(3,2) = f(2, f(3,1)) = f(2,2) = f(1, f(2,1)) = f(1,2) = 4$

$f(2,3) = f(1, f(2,2)) = f(1,4) = 6$

Name: .... **Matr.-Nr.: ....**

Aufgabe 6: **Persönlichkeiten** (2+2+2 Punkte)

Nennen Sie drei zentrale Persönlichkeiten der Informatik und beschreiben Sie jeweils kurz (höchstens drei Sätze) deren herausragende wissenschaftlichen Leistungen.

————Lösung————

Siehe Folien.

Name: .... **Matr.-Nr.: ....**

Aufgabe 7: **Fehlerkorrigierende Codes** (2+4 Punkte)

(a) Welche „Abstandsbedingung für Codewörter“ muss ein Code erfüllen, damit k-Bit-Fehler korrigiert werden können?

——————Lösung——————

Zwei beliebige Codewörter müssen einen Hamming-Abstand von mindestens $2k + 1$ haben. Somit ist der nächste Nachbar bei einem Codewort mit k Fehlern noch eindeutig bestimmt.

(b) Geben Sie eine binäre Codierung der vier Buchstaben a, b, c und d an, sodass:

i. Alle Codewörter genau Länge fünf haben.

ii. Fehler von genau einem Bit korrigiert werden können.

Zeigen Sie, dass die unter (a) genannte Bedingung auch für den von Ihnen entwickelten Code gilt.

——————Lösung——————

a	00000
b	11100
c	00111
d	11011

Hamming-Abstand des Codes ist 3 ($d_H(a,b) = d_H(a,c) = d_H(b,d) = d_H(c,d) = 3$ und $d_H(a,d) = d_H(b,c) = 4$), somit ist die Bedingung erfüllt.

Name: .... **Matr.-Nr.: ....**

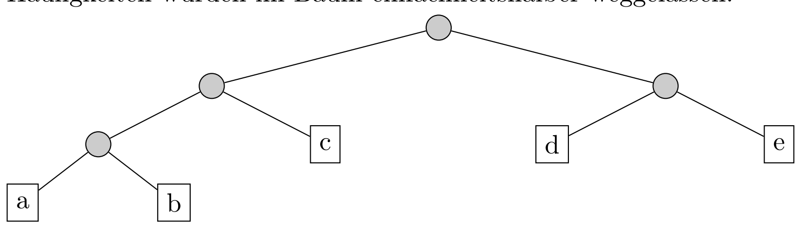
Aufgabe 8: **Huffman-Codierung** (2+4 Punkte)

Der Huffman-Baum zu einer gegebenen relativen Häufigkeit der einzelnen Zeichen wird wie folgt aufgebaut:

Erzeuge für jedes Zeichen x einen Teilbaum mit Knoten x und Gewicht $p_x :=$ relative Häufigkeit von x in M
while Es gibt mehr als einen Teilbaum **do**
  Suche zwei Teilbäume mit kleinsten Gewichten p_1 und p_2
  Vereinige die beiden Teilbäume zu einem mit Gewicht $p_1 + p_2$
end while

Gegeben folgende Häufigkeitsverteilung p für die Zeichen 'a', 'b', 'c', 'd', 'e' mit $p(a) = 1/12$, $p(b) = 1/12$, $p(c) = 1/6$, $p(d) = 1/4$, $p(e) = 5/12$.

(a) Gehört folgender Huffman-Baum zur gegebenen Häufigkeitsverteilung? Warum? Die Häufigkeiten wurden im Baum einfachheitshalber weggelassen.

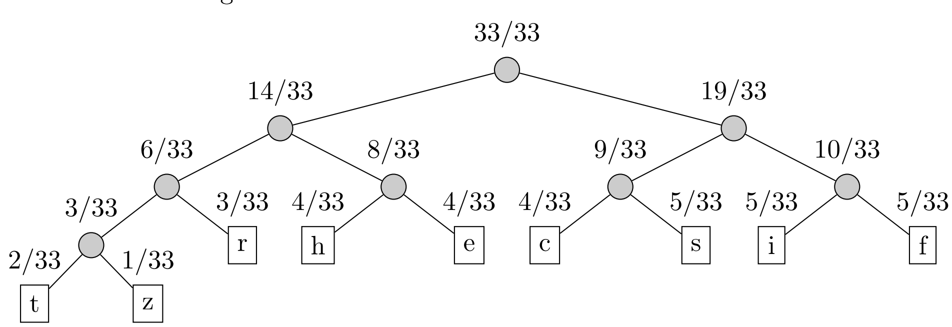


——————Lösung——————

Der Baum gehört aber nicht zur gegebenen Häufigkeit. Beispielsweise ist die relative Häufigkeit von 'a', 'b' und 'c' kleiner als die Häufigkeit von 'e'. Laut Huffman-Verfahren muss also der Teilbaum von 'a', 'b' und 'c' mit dem Teilbaum von 'd' zusammenfasst werden.

(b) Geben Sie den zur Zeichenkette "fischersfritzefischtfrischefische" zugehörigen Huffman-Baum an. Geben Sie die jeweiligen Gewichte der einzelnen Knoten an. Hinweis: Die betrachtete Zeichenkette hat 33 Zeichen.

——————Lösung——————



Name: .... **Matr.-Nr.: ....**

Aufgabe 9: **Zentrale Begriffe** (2+2+2+2 Punkte)

Zeigen Sie in jeweils maximal drei Sätzen Verknüpfungen der Informatik zwischen folgenden Begriffpaaren auf (ggf. mit Beispiel).

a) Zufall und Kryptologie

b) Zufall und algorithmische Komplexität

c) Abstraktion und Computational Thinking

d) Fehlerkorrigierende Codes und Datenkomprimierung

——————Lösung——————

Siehe Folien.