Klausur Grundlagen der Informatik Hochschule Ravensburg-Weingarten

Semester: AI2, WI2	**SS 2010**, 07.07.2010
Bearbeitungszeit: 90 Min.	90% Punkte entspr. Note 1,0
Hilfsmittel: kein prog. C	50% Punkte entspr. Note 4,0

Aufgabe 1 Umrechnung von Zahlensystemen

a) Geben Sie die binäre Zahl 1100101 in Dezimaldarstellung an. (1 Punkt)

b) Wie viele Stellen benötigt man zur binären Darstellung der größten Zahl, die 4-stellig noch hexadezimal dargestellt werden kann?
Geben Sie das Ergebnis mit Rechenweg an! (2 Punkte)

Aufgabe 2 Minimal aufspannende Bäume (MST)

a) Nennen Sie ein Kreissuchverfahren, das einen MST benötigt. (1 Punkt)

b) Nennen Sie einen berühmten Informatiker, der in Verbindung mit einem Algorithmus zur Findung des MST steht. (1 Punkt)

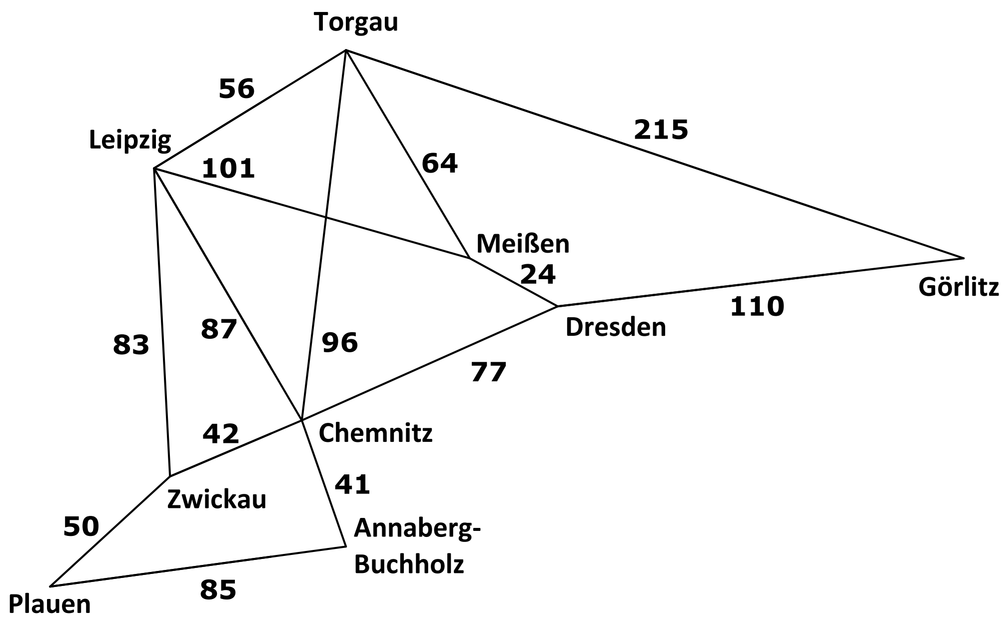
Aufgabe 2 (Fortsetzung) Minimal aufspannende Bäume (MST)

c) Übertragen Sie die Werte vom Graphen in die Entfernungstabelle (1 Punkt)

d) Warum darf die große, weiße Fläche in der Tabelle frei bleiben? (2 Punkte)

e) Warum muss in die dunkelgrauen Felder nichts eingetragen werden? (1 Punkt)

	Le	**To**	**Me**	**Gö**	**Dr**	**Ch**	**Zw**	**Pl**	**An**
Le									
To									
Me									
Gö									
Dr									
Ch									
Zw									
Pl									
An									



Aufgabe 2 (Fortsetzung) Minimal aufspannende Bäume (MST)

d) Ermitteln Sie mit einem geeigneten Algorithmus einen MST in diesem Graphen. Geben Sie die Menge der zugehörigen Kanten E_{MST} an. (5 Punkte)

e) Ist der Graph vollvernetzt? Begründen Sie anhand der Entfernungstabelle! (2 Punkte)

f) Ist der Graph planar? Begründen Sie! (2 Punkte)

Aufgabe 3 Problem des Handlungsreisenden (TSP)

a) Bestimmen Sie einen Pfad P_{TSP} als Lösung auf das TSP. (4 Punkte)

b) Welche Kanten haben P_{TSP} und E_{MST} gemeinsam? (2 Punkte)

Aufgabe 4 Asymptotik, Komplexität

a) Kreuzen Sie in der folgenden Tabelle **alle** zutreffenden Felder an, so dass die Landau'schen Symbole eine Verdeutlichung für die Schranke $g(n)$ zu $f(n)$ sind. Beispielsweise $f(n)=\omega(g(n))$ (8 Punkte)

$f(n)$	$g(n)$	O	o	Ω	ω	θ
n	$\ln(n)$					
n^2	$\ln(n)+n^2$					
$\log_4(n)$	$\log_3(n)$					
$3\ln(n!)$	$\lg(n^n)$					
$4n^2$	$16n+\cos(n^2)$					
$3n^2+n$	$5/(\cos n)^2$					

Punkte
1
1
1
1
2
2

b) Begründen Sie, dass gilt: $T\ n = \theta(2\lg(5n))$ (4 Punkte)

n	10	20	30	40	50	100
$T(n)$	56 s	66 s	72 s	76 s	79 s	s

c) Bestimmen Sie und tragen Sie T(100) in die Tabelle oben ein. (2 Punkte)

Aufgabe 5 Automaten / Sprachen / reguläre Ausdrücke

Gegeben sei die Grammatik G mit folgenden Produktionsregeln P, wobei mit der Variablen „S“ ∈ V = {S, A, B} begonnen werden muss:

P={ S → aA, A → bB, B → c|cA }

a) Geben Sie das kürzest mögliche Alphabet Σ der Sprache L(G) an. (2 Punkte)

b) Geben Sie die Grammatik als 4-Tupel an. (3 Punkte)

c) Geben Sie alle Wörter an, deren Länge 9 nicht übersteigt. (3 Punkte)

Aufgabe 5 (Fortsetzung) Automaten / Sprachen / reguläre Ausdrücke

d) Bestimmen Sie die Ableitung zum Wort „abcbc“! (3 Punkte)

e) Gehört das Wort „abcabc“ zu L(G)? (1 Punkt)

f) Konstruieren Sie einen Automaten, der alle Wörter von L(G) erkennt. (9 Punkte)

Beschreiben Sie ihn als 5-Tupel. Wählen Sie hierfür eine im Skript verwendete Darstellungsart der Zustandsübergänge, die Ihnen beliebt.

g) Geben Sie den regulären Ausdruck an, der genau L(G) erzeugt. (4 Punkte)

Aufgabe 6 Sortieralgorithmen

a) Geben Sie die Tiefe des Rekursionsbaumes bei QuickSort für den Best Case in Abhängigkeit von der Anzahl der Elemente „n" an. (3 Punkte)

b) Welcher Sortieralgorithmus bietet sich an, wenn Sie einen vorsortierten Datenbestand neu sortieren müssen, bei dem lediglich eine große Zahl hinten angesetzt wurde? (3 Punkte)

Aufgabe 7 Hashing

In eine Hashtabelle mit fünf Feldern sollen 50 Wörter verteilt werden. Verteilen Sie sie so, dass das **einfache uniforme Hashing** bestmöglich erfüllt ist! (2 Punkte)

Feldnummer	**Anzahl an Einträgen**
0	
1	
2	
3	
4	

b) Geben Sie die Auslastung α der Tabelle an. (2 Punkte)

c) Bedarf es hierbei der Kollisionsauflösung?

Begründen Sie! (3 Punkte)

Aufgabe 8 Komplexitätsberechnung eines Algorithmus

Gegeben ist ein Algorithmus mit folgendem Ablauf:

```
foo(n) {
      if (n > 1) {
            foo(n / 2);
            foo(n / 2);
      }
      bar();
      if (n > 1) {
            foo(n / 2);
            foo(n / 2);
      }
}
```

a) Markieren Sie alle rekursiven Aufrufe im obigen Quelltext. (1 Punkt)

b) Zeichnen Sie den Rekursionsbaum für foo(4). (3 Punkte)

c) Nutzen Sie die Kenntnis, dass der Algorithmus rekursiv ist (Mastertheorem!) zur Berechnung von foo(n), wobei $T_{bar}(n) = \theta\ (n^2)$ (6 Punkte)