Technische Universität Chemnitz
Professur Theoretische Informatik
Andreas Goerdt, Frank Schädlich

Chemnitz, den 12. Juni 2002

Wiederholungslausur Theoretische Informatik I

Alle Studiengänge

Wintersemester 2001/02

Hinweise

Die Aufgaben 1-8 sind von allen Klausurteilnehmern zu bearbeiten (150min Arbeitszeit). Prüfungsteilnehmer des Studienganges Mechatronik haben zusätzlich die Aufgaben 9 und 10 zu lösen (insgesamt 180min Arbeitszeit). Die Klausur ist bestanden, wenn mindestens 50% aller möglichen Punkte erzielt werden.

Es sind keine Hilfsmittel zugelassen. Alle Blätter sind zu numerieren sowie unbedingt mit Namen und Matrikelnummer zu versehen. Bitte vermerken Sie die Anzahl Ihrer Lösungsblätter auf der ersten Seite und verwenden Sie nur radierfeste Stifte in den Farben blau oder schwarz. Bitte verwenden Sie für eine neue Aufgabe ein neues Lösungsblatt an.

In manchen Aufgaben wird nur nach einer Laufzeit, einem Algorithmus, einer Heuristik, einer Eigenschaft o. ä. gefragt. Bitte geben Sie hier auch eine kurze Begründung für Ihre Antwort an.

Aufgabe 1: **2 + 2 Punkte**

$G = (V, E)$ sei ein gerichteter Graph, auf dem die Breitensuche durchgeführt wird. Die folgenden Aussagen beziehen sich auf den Zeitpunkt, zu dem die Breitensuche bei der Abarbeitung der Adjazenzliste von u auf den Knoten v stößt. Welche der folgenden Aussagen gelten?

a) Der Knoten v ist nie schwarz gefärbt.

b) Der Knoten v ist nie grau gefärbt

Aufgabe 2: **2 + 4 Punkte**

Eine Eingabe zum Problem des Handlungsreisenden sei durch eine $n \times n$-Matrix gegeben.

a) Wie viele mögliche Rundreisen kann es insgesamt geben?

b) Geben Sie die Laufzeit des besten Algorithmus der Vorlesung für dieses Problem an.

Aufgabe 3: **4 + 4 Punkte**

Betrachten Sie die Rekurrenz

$$\begin{aligned} T(1) &= 1 \\ T(n) &= 7 \cdot T\left(\frac{n}{2}\right) + n^2, \end{aligned}$$

wobei n eine Zweierpotenz ist.

a) Leiten Sie eine genaue explizite Formel (d. h. ohne Rekursionen, Summen- bzw. Produktzeichen o. ä.) für $T(n)$ her.

   Hinweis: Es bietet sich an, den Aufrufbaum zu verwenden.

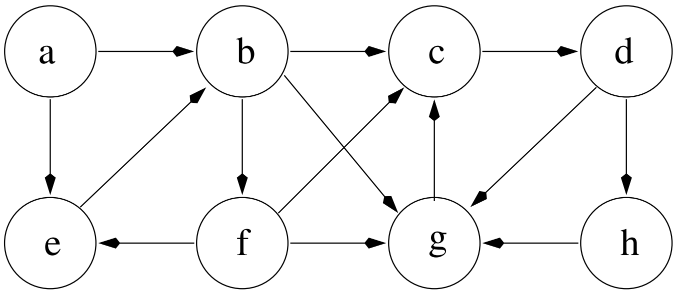
b) Beweisen Sie durch Induktion, daß für $T(n)$ die Formel aus a) gilt.

Aufgabe 4: **3 + 3 Punkte**

Zur Bestimmung von starken Zusammenhangskomponenten in gerichteten Graphen wird eine modifizierte Form der Tiefensuche eingesetzt.

a) Beschreiben Sie kurz die wesentlichen Punkte des Algorithmus zur Bestimmung von starken Zusammenhangskomponenten in gerichteten Graphen.

b) Die Adjazenzlisten der Knoten seien alphabetisch geordnet. Vollziehen Sie eine Tiefensuche, beginnend beim Knoten a, über dem nachfolgenden Graphen. Markieren Sie hierzu jeden Knoten des Graphen auf dem Lösungblatt in der Form Entdeckzeit/Beendezeit.



Geben Sie die starken Zusammenhangskomponenten des Graphen unter Verwendung der durchgeführten Tiefensuche an. In welcher Reihenfolge werden die starken Zusammenhangskomponenten gefunden?

Aufgabe 5: **4 + 2 Punkte**

a) Betrachten Sie den Algorithmus Bubble-Sort, der das Feld $A[1 \dots n]$ mit $A[i] = a_i$ sortiert:

```
FOR i:=1 to n-1
    FOR k=1 TO n-i
        IF (A[k+1] < A[k])
            tmp = A[k];
            A[k] = A[k+1];
            A[k+1] = tmp;
```

Stellen Sie für $n = 3$ den Entscheidungsbaum von Bubble-Sort auf. Füllen Sie die Blätter des Baumes mit der zugehörigen Sortierung des Feldes falls dieses Blatt erreicht werden kann.

b) Welcher Zusammenhang besteht zwischen dem Entscheidungsbaum und der Worst-Case-Laufzeit eines auf Vergleichen basierenden Sortierverfahrens?

Aufgabe 6: **4 Punkte**

Transformieren Sie

$$F = (\neg(a \wedge \neg b)) \vee (c \wedge d)$$

in eine erfüllbarkeitsäquivalente Formel in 3-KNF unter Verwendung des Polynomialzeitverfahrens der Vorlesung.

Aufgabe 7: **2 + 3 + 3 Punkte**

Es seien $v = a_1 \dots a_m$ und $w = b_1 \dots b_n$ Worte über einem gegebenen Alphabet. Es sei

$$\mathrm{lgT}(v, w) = \text{die Länge einer längsten gemeinsamen Teilfolge von } v \text{ und } w.$$

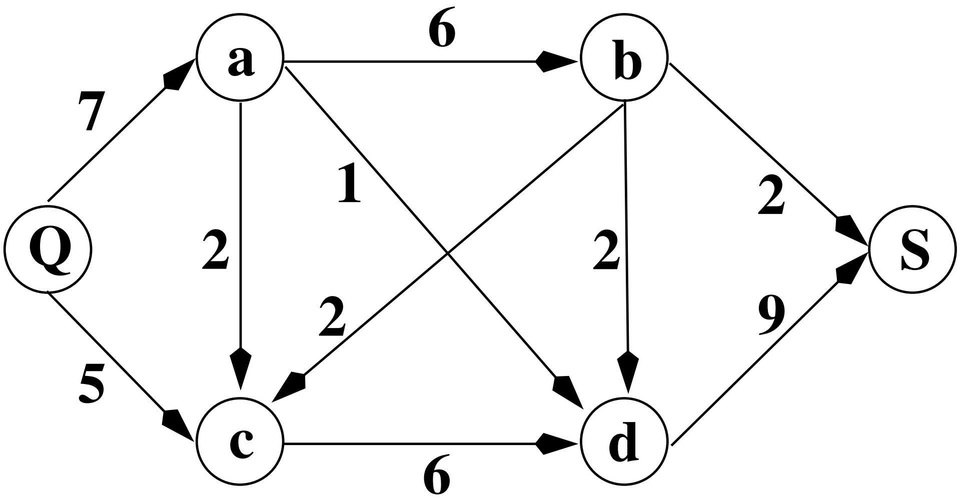
a) Geben Sie $\mathrm{lgT}(ab, bac)$ und $\mathrm{lgT}(aa, aba)$ an.

b) Geben Sie ein rekursives Gleichungssystem zur Berechnung von $\mathrm{lgT}(v, w)$ an.

c) Demonstrieren Sie die Arbeitsweise eines auf dem Prinzip des dynamischen Programmierens basierenden Algorithmus der Vorlesung, der bei $v = a_1 \dots a_m$ und $w = b_1 \dots b_n$ die Zeit $O(n \cdot m)$ benötigt. Füllen Sie die Tabelle auf dem Lösungsblatt für $v = abbc$ und $w = abcb$ entsprechend aus. Geben Sie die dadurch gefundene längste gemeinsame Teilfolge zusätzlich an.

Aufgabe 8 **4 Punkte**

Bestimmen Sie den maximalen Fluß im folgenden Netzwerk mit der Quelle Q und der Senke S unter Verwendung des Algorithmus von Ford-Fulkerson. Es ist auf dem Lösungsblatt der Fluß jeder Kante einzutragen. Geben Sie ebenfalls nach jeder Flußerhöhung den zugehörigen Restgraphen auf dem Lösungsblatt an.

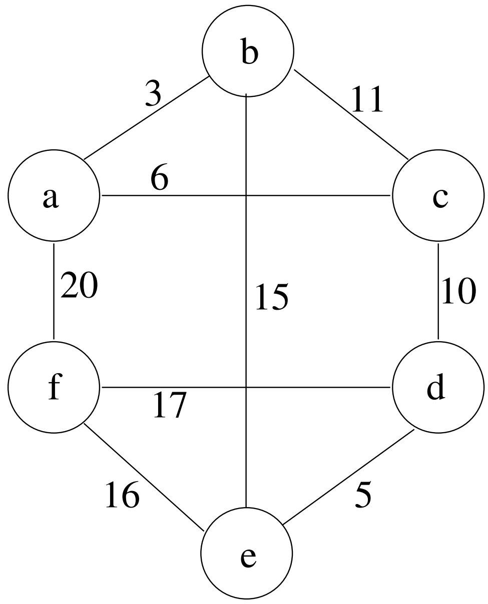


Aufgabe 9 (nur Studiengang Mechatronik): 5 Punkte

Gegeben sei ein Feld A, welches mit n ASCII-Zeichen (ganzzahliger 7-Bit-Wert) gefüllt ist. Diese Zeichen sollen in ein Feld B alphabetisch sortiert werden. Es steht ein Hilfsfeld C für maximal 512 Einträge zu Verfügung. Beschreiben Sie einen Algorithmus, der diese Sortierung effizient, d. h. im Worst-Case möglichst besser als $O(n \log n)$ bestimmt.

Aufgabe 10 (nur Studiengang Mechatronik): 3 + 2 Punkte

a) Bestimmen Sie im nachfolgenden ungerichteten und gewichteten Graphen den minimalen Spannbaum mit Hilfe des Kruskalalgorithmus und tragen Sie diesen Spannbaum im Lösungsblatt ein.



b) Im Kruskalalgorithmus der Vorlesung wird die Union-Find-Datenstruktur eingesetzt. Welche Heuristik stellt sicher, daß die Find-Operation die Laufzeit $O(\log n)$ hat (n sei dabei die maximale Anzahl von Elementen in der Datenstruktur)?