**KLAUSUR**

**zur Lehrveranstaltung "Algorithmen"**

Name: [redacted]

| | | |
|---|---|---|
| Klausur-Punkte: 73 / **75** | Übungs-Punkte: 25 / **25** | Vortrags-Punkte: / |
| Gesamt-Punkte: 98 / **100** | 1,0 | K. Schiele 18.7.07 |

**Aufgabe 1 (7 Punkte):**

7

Ein schlichter, zusammenhängender Graph **G = (V,E)** sei durch die folgenden Inzidenzlisten gegeben:

| | | | |
|---|---|---|---|
| $B_1$=[1,4,9,18], | $B_2$=[1,12,29], | $B_3$=[6,12,18,33], | $B_4$=[9,31], |
| $B_5$=[7,31], | $B_6$=[3,4,7,8], | $B_7$=[5,8,10,11], | $B_8$=[3,6,10,32], |
| $B_9$=[5,13,32,33], | $B_{10}$=[11,14,15,30], | $B_{11}$=[13,14,16,17], | $B_{12}$=[2,15], |
| $B_{13}$=[16,20], | $B_{14}$=[17,19,21,22], | $B_{15}$=[2,19,23,24], | $B_{16}$=[20,21,25,29], |
| $B_{17}$=[22,23,26], | $B_{18}$=[25,26,27], | $B_{19}$=[27,28,30], | $B_{20}$=[24,28]. |

**a)** Besitzt **G** einen Euler-Weg? nein

Begründen Sie kurz Ihre Antwort!

- Laut Satz 3 kann G nur einen Euler-Weg haben, wenn er keinen oder genau zwei ungerade Knoten hat
- G hat vier ungerade Knoten

**b)** Wenn **G** durch Adjazenzlisten dargestellt werden soll, wie lautet dann die Adjazenzliste $A_1$?

$A_1=[2,6,4,3]$

**Aufgabe 2 (13 Punkte):**

13

6 Dörfer in der Umgebung der Stadt Aachen sollen an die Erdgasversorgung angeschlossen werden, die in Aachen schon existiert.

Die Kosten für den Bau einer Erdgasleitung zwischen je 2 Orten werden auf folgende Beträge (in Tausend Euro) geschätzt:

| | Aachen | Brosen | Clarau | Dröve | Erlebach | Falkenau | Grüntal |
|---|---|---|---|---|---|---|---|
| **Aachen** | - | 15 ✓ | 20 | 28 | 12 ✓ | 22 | 35 |
| **Brosen** | 15 | - | 23 | 30 | 42 | 60 | 24 |
| **Clarau** | 20 | 23 | - | 10 | 33 | 8 | 6 |
| **Dröve** | 28 | 30 | 10 | - | 50 | 5 | 18 |
| **Erlebach** | 12 | 42 | 33 | 50 | - | 25 | 16 |
| **Falkenau** | 22 | 60 | 8 | 5 | 25 | - | 9 |
| **Grüntal** | 35 | 24 | 6 | 18 | 16 | 9 | - |

**a)** Das kostengünstigste Leitungssystem soll mittels des Algorithmus von Kruskal entworfen werden. Notieren Sie die ersten 4 „Kanten" dieses billigsten Leitungssystems in der Reihenfolge, in der sie der Algorithmus wählt!

| 1. | 2. | 3. | 4. |
|---|---|---|---|
| [D,F] | [C,G] | [C,F] | [A,E] |

**b)** Das kostengünstigste Leitungssystem soll mittels des Algorithmus von Prim entworfen werden. Notieren Sie die ersten 3 „Kanten", die der Algorithmus wählt, wenn die Stadt Aachen als Startknoten benutzt wird!

| 1. | 2. | 3. |
|---|---|---|
| [A,E] | [A,B] | [E,G] |

**Aufgabe 3 (15 Punkte):**

13

Ein schlichter Graph **G** mit 16 Knoten $\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_{16}}$ sei durch die folgenden Adjazenzlisten gegeben :

| | | | |
|---|---|---|---|
| $A_1$=[5,6], | $A_2$=[3,7,9], | $A_3$=[2,9,11,13,16], | $A_4$=[9,10], |
| $A_5$=[1,10,12], | $A_6$=[1,12], | $A_7$=[2,8,15], | $A_8$=[7,12,16], |
| $A_9$=[2,3,4], | $A_{10}$=[4,5,14], | $A_{11}$=[3,13], | $A_{12}$=[5,6,8], |
| $A_{13}$=[3,11], | $A_{14}$=[10,15], | $A_{15}$=[7,14], | $A_{16}$=[3,8]. |

Zur Bestimmung eines kürzesten Weges von $\mathbf{u_1}$ zu allen anderen Knoten von **G** soll das <u>BFS-Programm von Moore</u> (Skript, S. 42) verwendet werden.

Dokumentieren Sie den Beginn des Programmablaufs, indem Sie in die folgende Tabelle die ersten 14 Elemente in der Reihenfolge eintragen, in der Sie in die Warteschlange **Q** hinein gelangen, und die Markierungen **m**, die die betreffenden Knoten im Laufe des Algorithmus bekommen!

Beachten Sie dabei die in den Adjazenzlisten vorhandene Reihenfolge!

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Q** | 1 | 5 | 6 | 10 | 12 | 4 | 14 | 8 | 9 | 15 | 7 | 16 | 2 | 3 |
| **m** | 0 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 5 |

Ermitteln Sie auf der Grundlage dieses Ergebnisses mittels des rückverfolgenden Algorithmus (Skript, S. 42) den kürzesten Weg vom Knoten $\mathbf{u_1}$ zum Knoten $\mathbf{u_2}$!

Beachten Sie dabei die in den Adjazenzlisten vorhandene Reihenfolge!

Dokumentieren Sie den Ablauf des Algorithmus in der folgenden Tabelle!

| | | | | | | |
|---|---|---|---|---|---|---|
| **i** | 5 | 4 | 3 | 2 | 1 | 0 |
| $\mathbf{V_i}$ | $u_2$ | $u_9$ | $u_4$ | $u_{10}$ | $u_5$ | $u_1$ |

Fassen Sie den in der Tabelle ermittelten kürzesten Weg zusammen :

**W** = $u_1 u_5 u_{10} u_4 u_9 u_2$

15

**Aufgabe 4 (15 Punkte):**

Gegeben ist der paare Graph **G** in Bild 1 mit dem Matching

**M** = { $[a_1,b_2]$, $[a_2,b_1]$, $[a_3,b_5]$, $[a_5,b_4]$ }.

Wenden Sie die Ungarische Methode an, um die unten genannten Fragen zu beantworten!

Wählen Sie dabei im Falle mehrerer Auswahlmöglichkeiten immer den Knoten mit dem kleinsten Index!

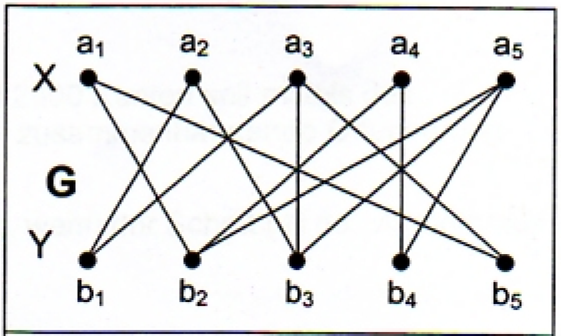
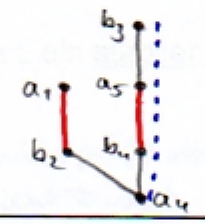



**Bild 1**

**a)** Dokumentieren Sie den Verlauf der Methode in der folgenden Tabelle!

| S | T | $\overset{?}{=}$ | P(S) | $y \in$ P(S) | y M-ges. | y M-unges. |
|---|---|---|---|---|---|---|
| $\{a_4\}$ | $\emptyset$ | $\neq$ | $\{b_2,b_4\}$ | $b_2$ | $a_1$ | |
| $\{a_1,a_4\}$ | $\{b_2\}$ | $\neq$ | $\{b_2,b_4,b_5\}$ | $b_4$ | $a_5$ | |
| $\{a_1,a_4,a_5\}$ | $\{b_2,b_4\}$ | $\neq$ | $\{b_2,b_3,b_4,b_5\}$ | $b_3$ | | $a_4 \to b_3$ |
| | | | | | | |

**b)** Zeichnen Sie den ersten **M**-alternierenden Baum, der dabei „wächst"!




**c)** Findet die Ungarische Methode einen **M**-erweiternden Weg in **G**? Wenn ja, geben Sie diesen an!

ja

$W = a_4 b_4 a_5 b_3$

**d)** Enthält **G** ein vollständiges Matching von **X** in **Y**? Wenn ja, notieren Sie dieses!

ja

$M = \{[a_1,b_2],[a_2,b_1],[a_3,b_5],[a_4,b_4],[a_5,b_3]\}$

**Aufgabe 5 (3 Punkte):**

Ein orientierbarer Graph **G** mit 210 Knoten und 2500 Kanten soll mittels des <u>Algorithmus von Hopcroft und Tarjan</u> eine stark zusammenhängende Orientierung erhalten. 3

Wie viele Kanten von **G** besitzen eine Richtung, wenn der Schritt (4) des Algorithmus (Skript Seite 66) erreicht wird?

| 209 |
|---|

M = V -> 210 Knoten markiert
L> Gerüst -> n-1 Kanten haben eine Richtung

**Aufgabe 6 (7 Punkte):**

Betrachten Sie den folgenden Pseudocode eines Sortieralgorithmus: 7

```
a[0..n-1] - zu sortierende Eingabedaten-Reihung mit n Elementen
i und j   - Laufindizes

klausurSort (a[0],...,a[n-1])
for i=n-1 downto 1 do
  for j=0 to i-1 do
    if a[j] > a[j+1] then
        tausche a[j] und a[j+1]
    end if
  end for
end for
```

**a)** Ist der Algorithmus `klausurSort` ein <u>stabiler Algorithmus</u>?

| ja |
|---|

Begründen Sie Ihre Antwort!

klausurSort ist stabil, da er nicht prüft ob a[j] auch gleich a[j+1] ist und diese Elemente dann auch nicht tauscht.
Die relative Reihenfolge der Elemente bleibt also erhalten.

**b)** Welche Eingabedaten bewirken eine best-case-Laufzeit dieses Algorithmus?

Ein aufsteigend sortiertes Array würde eine best-case-Laufzeit bewirken, da keine Tauschoperationen ausgeführt werden.

**c)** Welche Eingabedaten bewirken eine worst-case-Laufzeit dieses Algorithmus?

Ein absteigend sortiertes Array würde eine worst-case-Laufzeit bewirken, da immer getauscht wird und große Elemente von vorn bis nach hinten durchgereicht werden.

**Aufgabe 7 (7 Punkte):**

Die folgende Eingabefolge (A) soll aufsteigend sortiert werden:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 18 | 14 | 11 | 7 | 12 | 4 | 8 | 5 |

7

**a)** Der <u>QuickSort-Algorithmus</u> soll dafür verwendet werden, wobei das <u>letzte Element</u> der Folge <u>als Trennelement</u> dienen soll.

Notieren Sie die Eingabefolge nach der ersten Vertauschung von 2 Elementen, und begründen Sie, weshalb diese Vertauschung stattfindet!

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 14 | 11 | 7 | 12 | 18 | 8 | 5 |

Das erste Mal wird 18 mit 4 vertauscht. 4 ist kleiner bzw. gleich dem Pivotelement und 18 ist größer bzw. gleich dem Pivotelement. Der rechte Zeiger bleibt also auf der ~~w~~ 4 stehen, der Linke auf der 18, und beide Elemente werden getauscht, da sie sich in der jeweils „falschen" Teilfolge befinden.

**b)** Die Eingabefolge (A) soll mittels des <u>HeapSort-Algorithmus</u> sortiert werden.

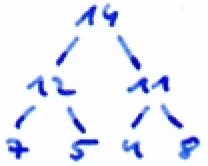
Notieren Sie die Eingabefolge nach der ersten Vertauschung von 2 Elementen, und begründen Sie, weshalb diese Vertauschung stattfindet!

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 14 | 12 | 11 | 7 | 5 | 4 | 8 | 18 |

Zuerst wird ein Max-Heap erzeugt. ~~Dabei werden hier nur die 8 und 4 getauscht~~ Dieser ist hier schon gegeben. Die Wurzel des Baumes wird dann mit dem letzten Blatt getauscht. Die Wurzel wird an das Ende des Arrays (Ergebnisarray) geschrieben. Danach wird wieder die Max-Heap-Eigenschaft für jedes Element geprüft und wiederhergestellt.

(✓)

Somit ergibt sich folgender Max-Heap:



**Aufgabe 8 (8 Punkte):**

8

Die Funktionen f und g seien definiert durch:

$$f(n) = 10n^4 - 12n^3 + 14$$

$$g(n) = 200n^3 - n$$

Sind die folgenden Aussagen wahr oder falsch?

| | w / f |
|---|---|
| $g(n) \in O(f(n))$ | w |
| $f(n) \in O(g(n))$ | f |
| $f(n) \in O(n^4)$ | w |
| $g(n) \in O(n^2)$ | f |

**Achtung:**
Für jede falsche Antwort werden 2 Punkte abgezogen.