

Prof. Petra Mutzel  
Günther Raidl  
Gunnar Klau  
Gabriele Kodydek  
René Weiskircher

Wintersemester 2001/2002

**Prüfung zur Vorlesung  
Algorithmen und Datenstrukturen 2  
Januar 2002**

a) Machen Sie bitte die folgenden Angaben in deutlicher Blockschrift:

Vorname: \_\_\_\_\_ Nachname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_ Studienkennzahl: \_\_\_\_\_

b) Klausur soll gewertet werden für (nur ein Kreuz!):

- ☐ VO-Prüfung Algorithmen und Datenstrukturen 2 (*default*)
- ☐ LU-Test Algorithmen und Datenstrukturen 2
- ☐ VO-Prüfung **und** LU-Test Algorithmen und Datenstrukturen 2

c) Legen Sie während des Tests Ihren Studentenausweis vor sich auf das Pult.

d) Schreiben Sie die Lösungen direkt auf das jeweilige Aufgabenblatt. Wenn Ihnen das Papier ausgeht, bitten Sie die Aufsicht um Nachschub. Es ist nicht erlaubt, eigenes Papier zu verwenden!

e) Denken Sie daran, dass keinerlei Hilfsmittel erlaubt sind – weder Taschenrechner, irgendwelche Unterlagen, Mobiltelefone etc.

**VOR DER ABGABE AUSZUFÜLLEN:**

Geben Sie bitte die Anzahl der zusätzlich abgegebenen Blätter an: \_\_\_\_\_

**Resultat:**

Aufgabe	A 1	A 2	A 3	A 4	A5	
maximale Punktzahl	10	10	10	10	10	50
erreicht						

Viel Erfolg!

### Aufgabe 1.A: Tiefensuche

(10 Punkte)

Gegeben ist eine Datenstruktur *Stack*, die folgende Operationen zulässt:

- *Push*: Fügt ein Objekt in den Stack ein
- *Pop*: Entfernt das Objekt aus dem Stack, das als letztes eingefügt wurde und liefert es zurück

Benutzen Sie den Stack um eine nicht rekursive Version von Tiefensuche (Depth First Search) zu entwerfen. Beschreiben Sie erst in wenigen Worten den Ablauf Ihres Algorithmus und geben Sie dann den Algorithmus in Pseudocode an. Den Stack können Sie dabei als “Black Box” betrachten, d.h. sie können ihn benutzen, ohne seine Funktionsweise zu beschreiben.

**Aufgabe 2.A: Simulated Annealing****(10 Punkte)**

Beim Bin-Packing-Problem ist eine Menge  $U = \{u_1, \dots, u_n\}$  von Gegenständen gegeben, wobei Gegenstand  $u_i$  ein Gewicht von  $g_i$  Einheiten hat für  $1 \leq i \leq n$ . Die Aufgabe ist nun, diese Gegenstände in eine möglichst kleine Anzahl von Behältern zu packen, die jeweils maximal ein Gewicht von  $B$  Einheiten aufnehmen können. Eine Lösung besteht aus einer Liste, die für jeden benutzten Behälter die Gegenstände im Behälter angibt. Dieses Problem soll nun mittels “Simulated Annealing” gelöst werden. Dazu muss eine Nachbarschaftsfunktion angegeben werden.

- a) (4 Punkte)  
Beschreiben Sie eine geeignete Nachbarschaftsfunktion, die aus einer gültigen Lösung eine ähnliche gültige Lösung berechnet.
- b) (6 Punkte)  
Begründen Sie, warum Ihre Nachbarschaftsfunktion die Bedingungen “Erreichbarkeit des Optimums” und “Lokalitätsbedingung” erfüllt.

### Aufgabe 3.A: Geometrische Algorithmen

(10 Punkte)

Betrachten Sie das Problem des Schnittes von allgemeinen Liniensegmenten: Gegeben sei eine Menge von  $n$  Liniensegmenten in der Ebene in allgemeiner Lage; gesucht werden alle echten Schnittpunkte zwischen den Liniensegmenten. Dieses Problem kann mit Hilfe eines Sweep-Line-Algorithmus effizient gelöst werden.

a) (3 Punkte)

Wieviele Schnittpunkte gibt es maximal? Wie groß ist der Worst-Case-Aufwand des Sweep-Line-Algorithmus in  $\Theta$ -Notation? Geben Sie ein Beispiel für den Worst-Case mit  $n = 4$  an.

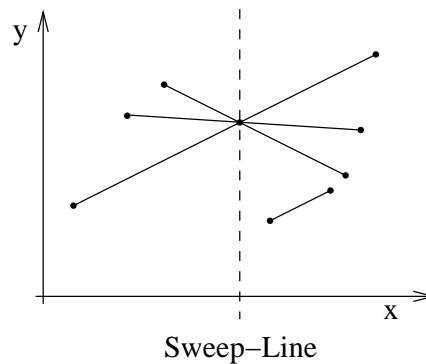
b) (4 Punkte)

Beschreiben Sie die sogenannte *Sweep-Status-Struktur* (*SSS*), die im Rahmen des Sweep-Line-Algorithmus für das Segment-Schnitt-Problem in allgemeiner Lage verwendet wird. Beantworten Sie dabei insbesondere folgende Fragen: Welche Daten werden in *SSS* gespeichert? Welche Operationen stellt *SSS* zur Verfügung? Durch welche Datenstruktur kann *SSS* effizient realisiert werden? Welchen Aufwand haben die Operationen auf *SSS*?

c) (3 Punkte)

Bei der Realisierung des Sweep-Line-Algorithmus für das Segment-Schnitt-Problem gehen wir von der vereinfachenden Annahme der *allgemeinen Lage* der Segmente aus. In der Praxis kann es aber beispielsweise vorkommen, dass sich mehrere Segmente in einem Punkt schneiden.

Wie verhält sich der Sweep-Line-Algorithmus in dem unten dargestellten Sonderfall, bei dem drei Liniensegmente einander an einem Punkt schneiden?



**Aufgabe 4.A: Suchen in Texten – Tries****(10 Punkte)**

Gegeben sei ein Alphabet  $\Sigma = \{ 'A', 'B', 'C', 'D', 'E' \}$ . Zeichnen Sie einen Indexed Trie, der die Worte  $\{AB, BAB, BC, BE, DA, DC, DE\}$  beinhaltet.

Führen Sie anschließend Suffix Compression in dem von Ihnen gezeichneten Indexed Trie durch. Kennzeichnen Sie die Änderungen deutlich!

Aus dem resultierenden Indexed Trie mit Suffix Compression soll nun ein Packed Trie erstellt werden. Verwenden Sie dazu die Greedy-Heuristik aus der Vorlesung bzw. aus dem Skriptum. Zeigen Sie dabei mit Hilfe einer kleinen Graphik, wie in der Vorlesung bzw. im Skriptum gezeigt, auf welche Weise die Knoten gepackt werden und zeichnen Sie den Packed Trie.

**Aufgabe 5.A: Miller-Rabin-Primzahltest****(10 Punkte)**

a) (5 Punkte)

Beim Miller-Rabin-Primzahltest wird eine Funktion benutzt, die für zwei positive ganze Zahlen  $a$  und  $n$  prüft, ob

$$a^{n-1} \equiv 1 \pmod{n} \quad (1)$$

gilt. Nehmen Sie an, dass die Binärdarstellung von  $n-1$  in der Form  $(b_{k-1}, \dots, b_0)$  gegeben ist. Geben Sie Pseudo-Code für eine Funktion “Zeuge( $a, n$ )” an, die  $a$  und  $n$  auf die Eigenschaft (1) testet. Ihre Funktion soll Laufzeit  $O(k^3)$  haben, wobei davon ausgegangen wird, dass die Multiplikation zweier  $k$ -bit Zahlen Komplexität  $O(k^2)$  hat.

b) (5 Punkte)

Führen Sie Ihren Algorithmus für  $a = 7$  und  $n = 117$  durch. Geben Sie für jeden Schleifendurchlauf den Wert Ihrer Variablen an.







