

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 Glück!

Aufgabe 1.A: Minimale aufspannende Bäume**(10 Punkte)**

Gegeben sind zwei Graphen G_1 und G_2 mit jeweils einer Million Knoten. Graph G_1 hat vier Millionen Kanten, während G_2 250 Milliarden Kanten hat. Die Kanten in beiden Graphen haben ganzzahlige Kosten. Sie wollen nun in beiden Graphen jeweils einen aufspannenden Baum mit minimalen Kosten berechnen. Welchen Algorithmus benutzen Sie für G_1 und welchen für G_2 , um möglichst kurze Laufzeiten zu erreichen? Begründen Sie Ihre Antwort, indem sie die Laufzeiten der von Ihnen verwendeten Algorithmen angeben. Dabei sollten Sie auch angeben, warum die Algorithmen die von Ihnen angegebenen Laufzeiten aufweisen (kurze Beschreibung der Algorithmen und der Funktionsweise und Eigenschaften der verwendeten Datenstrukturen).

Aufgabe 2.A: Subset-Sum-Problem**(10 Punkte)**

Eine Instanz des Subset-Sum-Problems besteht aus einer Menge A von n ganzen positiven Zahlen und einer oberen Schranke T . Wir bezeichnen eine Teilmenge A' von A als *zulässig*, wenn die Summe aller Elemente von A' höchstens T ist. Die Summe der Elemente in einer zulässigen Teilmenge bezeichnen wir auch als den *Wert* der Teilmenge. Wir suchen eine zulässige Teilmenge von A mit einem möglichst großen Wert (im Idealfall T). Im Folgenden nehmen wir an, dass für ein gegebenes Subset-Sum-Problem eine zulässige Teilmenge von A mit Wert größer oder gleich $\frac{T}{2}$ existiert.

- a) (4 Punkte)
Geben Sie Pseudo-Code für einen polynomiellen Algorithmus an, der eine zulässige Teilmenge mit einem Wert von mindestens $\frac{T}{2}$ berechnet.
- b) (2 Punkte)
Geben Sie die Laufzeit Ihres Algorithmus in Θ -Notation an und begründen Sie diese Laufzeit.
- c) (4 Punkte)
Beweisen Sie, dass Ihr Algorithmus eine Lösung mit einem Wert von mindestens $\frac{T}{2}$ berechnet.

Aufgabe 3.B: 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)

Wie groß ist der Aufwand des Sweep-Line-Algorithmus (in Θ -Notation), um festzustellen, dass es in einer gegebenen Menge von Liniensegmenten *keinen* Schnittpunkt gibt? Begründen Sie Ihre Antwort mit wenigen Worten.

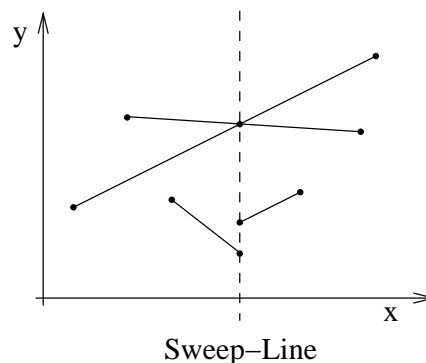
b) (4 Punkte)

Beschreiben Sie die sogenannte *Ereignisdatenstruktur* (ES), 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 ES gespeichert? Wieviele Einträge hat ES mindestens, wieviele maximal? Welche Operationen stellt ES zur Verfügung? Durch welche Datenstruktur kann ES effizient realisiert werden? Welchen Aufwand haben die Operationen auf ES ?

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 die Endpunkte und Schnittpunkte der Liniensegmente nicht paarweise verschiedene x -Koordinaten haben.

Wie verhält sich der Sweep-Line-Algorithmus in dem unten dargestellten Sonderfall, bei dem an einer x -Koordinate zwei Endpunkte sowie ein Schnittpunkt aufeinander treffen?



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

Gegeben sei ein Alphabet $\Sigma = \{ 'A', 'B', 'C', 'D', 'E' \}$. Zeichnen Sie einen Indexed Trie, der die Worte $\{AB, AD, C, DA, DB, DCA, EA\}$ 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: Randomisierte Algorithmen – Primzahltest

(10 Punkte)

a) (3 Punkte)

Ein naiver Ansatz zur Beantwortung der Frage, ob eine Zahl n eine Primzahl ist, ist die Divisionsmethode, die in der Vorlesung bzw. im Skriptum vorgestellt wurde. Erläutern Sie, warum diese Methode für große Zahlen nicht praktikabel ist.

b) (4 Punkte)

Erklären Sie die Idee des Algorithmus von Miller-Rabin zum Test der Primalität der Zahl n mit wenigen Worten. Liefert dieser Algorithmus exakte Ergebnisse?

c) (3 Punkte)

Was versteht man in diesem Zusammenhang unter *Pseudoprimzahlen*?

