Algorithmen und Programmieren 3 **Matrikelnummer:** ______________

Nachklausur zur Endklausur **FAMILIENNAME:** ______________

WS 2003/2004. Dienstag, 6. April 2004. **Vorname:** ______________

Anleitung: Streichen Sie *eine* Aufgabe deutlich auf dem Angabeblatt. Diese Aufgabe wird nicht in die Bewertung einbezogen. Wenn Sie selbst keine Aufgabe streichen, wird die erste Aufgabe nicht in die Bewertung einbezogen. Bearbeiten Sie die übrigen drei Aufgaben.

Bearbeiten Sie jede Aufgabe auf einem getrennten Blatt. Jede Aufgabe hat 10 Punkte. Bearbeitungszeit: 90 Minuten

Aufgabe	1	2	3	4	Summe
Punkte	10	10	10	10	30
Punkte					

---

1. Die sogenannte *Erdős-Zahl*[1] ist nach dem berühmten ungarischen Mathematiker Paul Erdős (1913–1996) benannt. Er selbst hat Erdős-Zahl 0. Alle seine Koautoren, mit denen er wissenschaftliche Arbeiten publiziert hat, haben Erdős-Zahl 1. Wer eine gemeinsame Arbeit mit jemandem mit Erdős-Zahl 1 geschrieben hat, aber nicht mit Erdős selbst, hat Erdős-Zahl 2, und so weiter. Personen, die nicht auf diese Weise mit Erdős verbunden sind, haben Erdős-Zahl ∞.

   Der *Kollaborationsgraph* enthält alle lebenden oder toten Personen als Knoten. Zwei Knoten sind durch eine Kante verbunden, wenn die beiden Personen gemeinsame Autoren einer wissenschaftlichen Publikation sind.

   (a) Geben Sie einen *Algorithmus* an, der für eine gegebene Person im Kollaborationsgraphen die Erdős-Zahl berechnet.

   (b) Geben Sie einen Algorithmus an, der die größte endliche Erdős-Zahl bestimmt.

   Algorithmen aus der Vorlesung können Sie direkt verwenden; die müssen Sie nicht "ausprogrammieren".

2. (a) (5 Punkte) Zeichne sie den

      i. digitalen Suchbaum

      ii. den komprimierten digitalen Suchbaum

   für die Bitketten `0011000101`, `001111001`, `01011`, `00111111`, `01010`, `001101`. Zeigen Sie, wie die Bäume nach dem Einfügen des Wertes `0011000` aussehen. Entfernen Sie `001111001` und zeichnen Sie die Bäume danach.

   (b) (5 Punkte) Wie kann man aus der Verschiebefunktion des Wortes $x\#w\$$ berechnen, ob x ein Teilwort von w ist? (Hier sind $\#$ und $\$$ neue Buchstaben, die nicht in x und w vorkommen.) *Begründen Sie Ihre Antwort!*

[1] `http://personalwebs.oakland.edu/~grossman/erdoshp.html`

3. Eine stückweise lineare Funktionen $f\colon (u,v] \to \mathbb{R}$ ist durch eine zusammenhängende Folge von Intervallen mit entsprechenden linearen Funktionen $f(x) = ax + b$ für jedes Intervall gegeben, z. B.

$$f(x) = \begin{cases} -x, & \text{für } -1 < x \leq 0 \\ 2x, & \text{für } 0 < x \leq 2 \\ -3x+9, & \text{für } 2 < x \leq 4 \end{cases}$$

Die Datenstruktur soll folgende Operationen erlauben:

(a) Erzeugen einer Funktion $f(x) = ax + b$ mit einem einzigen Intervall $(u, v]$.

(b) Berechnung des Wertes einer solche Funktion $f(x)$ an einer gegebenen Stelle x.

(c) Addition zweier stückweise linearer Funktionen.

Nehmen Sie der Einfachheit halber an, dass alle Intervalle halboffen sind, und zwar links offen und rechts abgeschlossen, wie im obigen Beispiel.

Der Datentyp ist durch die folgenden Axiome (mit geeigneten Signaturen) algebraisch spezifiziert:

$$\begin{aligned} \mathit{berechne}(\mathit{stLinFun}(u,v,a,b),x) &= \begin{cases} ax+b, & \text{für } u \leq x < v \\ 0, & \text{sonst} \end{cases} \\ \mathit{berechne}(\mathit{add}(f,g),x) &= \mathit{berechne}(f,x) + \mathit{berechne}(g,x) \end{aligned}$$

Implementieren Sie diesen Datentyp in Java oder Haskell und beweisen Sie die Korrektheit.

4. Bestimmen Sie die untere konvexe Hülle der folgenden sortierten Punktmenge $P_1, \ldots, P_9$ mit dem inkrementellen Algorithmus aus der Vorlesung. Geben Sie die Folge der Orientierungstests an, die der Algorithmus ausführt.