Technische Universität Berlin **SS 2013**
Fakultät II, Institut für Mathematik
Sekretariat MA 5–2, Frau Kiefer

Prof. Dr. Martin Skutella
Torsten Gellert
Martin Groß
Dr. Max Klimm

Aylin Acikel, Katharina Bütow, Christian Döblin, Alexander Hopp, Daniel Kuske, Olivia Röhrig, Robert Rudow, Daniel Schmand, Hendrik Schrezenmaier, Mona Setje-Eilers, Judith Simon, Sebastian Spies, Fabian Wegscheider, Jan Zur

4. Übungsblatt Computerorientierte Mathematik II

Abgabe: 07.05.2013 (bis 14:15 Uhr in MA 004)

Bitte denkt daran, dass wir im Zweifel immer eine Begründung für eure Antworten haben möchten, ohne dass das jedes Mal explizit in der Aufgabenstellung steht.

1. Aufgabe (0 + 0 + 0 + 3 Punkte)

Gegeben sei eine Datei, deren Analyse ergeben hat, dass sie die in der unten stehenden Tabelle aufgeführten Zeichen mit den dort genannten Häufigkeiten enthält.

Zeichen	a	b	c	d	e	f	g	h
Häufigkeit	1	1	2	3	5	8	13	21

(a) Berechnet die Länge in Bits, die eine binäre Darstellung der Datei mit *fester*, d.h. konstanter Codierungslänge für jedes Zeichen mindestens benötigt.

(b) Konstruiert einen binären Baum mit Hilfe des Algorithmus von Huffman, mit dem ihr den Zeichen entsprechend ihrer Häufigkeiten unterschiedlich lange Codewörter zuordnet. Gebt anschließend die Codierung für jedes Zeichen an.

(c) Wieviel Speicherplatz (in Bits) spart ihr durch Verwendung dieses Huffman-Codes gegenüber der Codierung mit fester Länge aus (a), wenn ihr den Inhalt der oben genannten Datei ohne die Codetabelle abspeichert?

(d) Wie lang kann ein Codewort eines Huffman-Baums für n Zeichen maximal sein? Gebt eine Häufigkeitsverteilung an, mit der diese Schranke erreicht wird.

2. Aufgabe (6 Punkte)

Gegeben sei ein digitaler Text mit n verschiedenen Zeichen und ein binärer Baum, der einen zu diesem Text gehörigen Huffman-Code repräsentiert. Die Häufigkeit eines Zeichens ist zusammen mit dem Zeichen im Knoten abgespeichert. Entwerft eine rekursive Methode zur Berechnung der Länge des komprimierten Textes, zeigt ihre Korrektheit und gebt ihre Laufzeit in Θ-Notation an (mit Beweis).

3. Aufgabe (3 + 5 Punkte)

Ihr habt kennengelernt, wie man einen Binärbaum für einen bekannten Text erstellt, der die Information zur Huffman-Kodierung dieses Textes beinhaltet. In dieser Aufgabe nehmen wir an, dass wir nicht den Text sondern nur die Auftritts*wahrscheinlichkeiten* kennen. Der Algorithmus arbeitet dann mit diesen Wahrscheinlichkeiten anstatt den Häufigkeiten. Der resultierende Huffman-Code minimiert somit nicht die komprimierte (Datei-)Länge, sondern die *erwartete Codelänge*.

Betrachtet $n \geq 2$ Zeichen, die mit den Wahrscheinlichkeiten $p_1 \geq p_2 \geq \ldots \geq p_n$ auftreten. Dabei sei $p_i = 2^{-k_i}$, $k_i \in \mathbb{N}$ und $\sum_{i=1}^{n} p_i = 1$.

(a) Beweist, dass $p_{n-1} = p_n$.

(b) Zeigt formal, dass für den resultierenden Huffman-Code die erwartete Codelänge bei dieser Verteilung

$$-\sum_{i=1}^{n} p_i \log p_i$$

beträgt. *Anmerkung:* das ist bestmöglich für die erwartete Codelänge.

4. Aufgabe (2 + 1 Punkte)

Der American Standard Code for Information Interchange, kurz ASCII, ist eine Zeichenkodierung *fester* Länge, mit der 128 verschiedene Schrift- und Steuerzeichen dargestellt werden können. Eine bestimmte Datei zum Testen des ASCII-Codes enthalte jedes der 128 Zeichen genau einmal.

(a) Welche Struktur hat der Huffman-Baum, der beim zeichenweisen Codieren dieser Datei entsteht? Beweist eure Aussage.

(b) Wie verhält sich bei dieser Datei die Länge des Huffman-Codes (ohne die Codetabelle) zur Codierung mit ASCII?