Datenkompression (Rote), SS 2003, 6. Übungsblatt
Abgabe: bis Montag, 2. 6. 2003

23. (5 Punkte) Beweisen Sie folgenden Hilfssatz: Gegeben ist eine Folge von n Zahlen a_i der Form 2^{-l} mit $l \geq 0$, mit Summe A. Für jede natürliche Zahl $b \leq A$ gibt es eine Teilmenge $I \subseteq \{1, \ldots, n\}$ mit Summe $\sum_{i \in I} a_i = b$.

24. (10 Punkte) Vergleichen Sie die Kompressionsrate des adaptiven Huffman-Kodes und eines statischen Huffman-Kodes für folgende Eingaben:

    (a) $a \ldots ab \ldots bc \ldots cd \ldots d$
    (b) $a \ldots ab \ldots bc \ldots cd \ldots de$
    (c) $a \ldots ab \ldots bc \ldots c \cdots g \ldots gh \ldots hi$

    Es handelt sich dabei um vier beziehungsweise acht gleich lange Blöcke von gleichen Buchstaben, denen im Fall (b) und (c) ein einzelnes Symbol folgt.

    Bestimmen Sie die durchschnittlich benötigte Zahl von Bits pro Eingabezeichen, wenn die Länge der Nachricht gegen unendlich geht.

    Zusatzfrage (0 Punkte). Versuchen Sie, große Beispiele zu finden, bei dem das Längenverhältnis zwischen adaptiver und statischer Huffman-Kodierung möglichst groß und möglichst klein ist. Suchen Sie auch Beispiele wo der Unterschied zwischen adaptiver und statischer Huffman-Kodierung bezogen auf die mittlere Anzahl der Bits pro Eingabesymbol möglichst groß ist, und zwar in beiden Richtungen.

25. (0 Punkte) Beweisen Sie: In einem Baum, der vom Huffman-Algorithmus erzeugt worden ist, haben die vier Kinder von zwei inneren Knoten mit gleichem Gewicht ebenfalls alle das gleiche Gewicht. (Stimmt diese Aussage auch, wenn es Knoten vom Gewicht 0 gibt?)

26. Experimente mit Huffman-Kompression (Programmieraufgabe, 10 Punkte)
    Wenden Sie Ihr Programm aus Aufgabe 17 auf die sechs Dateien `abracadabra`, `book1`, `4letter.words`, `hw2p3b.wav`, `paper1`, und `progc` in

    `http://page.inf.fu-berlin.de/~rote/Lere/2003-SS/Datenkompression/Daten/`

    an. Bestimmen Sie die Länge jeder entstehenden komprimierten Datei in Bits insgesamt und in Bits pro Symbol der Eingabedatei. Schlüsseln Sie die entstehenden Gesamtlängen folgendermaßen auf:

    (a) Die Bits für den Huffman-Kode (sofern Sie keinen adaptiven Kode gewählt haben.)
    (b) Gegebenenfalls die Bits für die Übertragung des Alphabets (entweder als Blätter der Kode-Baumes oder als "neue" Symbole beim adaptiven Kodieren)
    (c) Die Bits für kodierte Nachricht, die aus den Wörter des (adaptiven) Huffman-Kodes selbst besteht.
    (d) Eventuelle zuzätzliche Bits, zum Beispiel zur Speicherung der Dateilänge oder zum Auffüllen auf volle Bytes.

27. (5 Punkte) Runden bei der arithmetischen Kodierung.
    Nehmen wir an, bei der arithmetischen Kodierung arbeiten wir mit ganzzahligen Intervallgrenzen. Wir haben zwei mögliche Symbole mit Wahrscheinlichkeiten $p_0 = 0{,}5555$ und $p_1 = 0{,}4445$ zu kodieren. Wie sollen wir in diesem Fall ein Intervall der Länge $L = 1000$ in zwei ganzzahlige Intervalle zerlegen, sodass die mittlere erwartete Nachrichtenlänge möglichst klein wird, wenn wir annehmen, dass p_0 und p_1 die tatsächlichen Wahrscheinlichkeiten für das Auftreten der Symbole sind.