64-041 Übung Rechnerstrukturen und Betriebssysteme

Aufgabenblatt 6 Ausgabe: 20.11., Abgabe: 27.11. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 6.1 (Punkte 5+5+10+10)

Logische- und Shift-Operationen: Realisieren Sie die folgenden Funktionen als *straightline*-Code in Java, das heißt ohne Schleifen, If-Else Abfragen bzw. ternären Operator `..  ?  ..  :  ...` Außerdem dürfen nur einige der logischen und arithmetischen Operatoren benutzt werden:

```
! ~ & ^ | + << >> >>>
```

Alle Eingabeparameter und Rückgabewerte sind jeweils (32-bit) Integerwerte.

(a) `bitNor(x,y)` Diese Funktion soll das bitweise NOR liefern: $\overline{x_i \vee y_i}$. Als Operatoren dürfen nur `&` und ~ (AND, Negation) benutzt werden.

(b) `bitXor(x,y)` Diese Funktion soll die XOR-Verknüpfung (Antivalenz) realisieren: $x_i \neq y_i$. Als Operatoren dürfen nur `&` und ~ (AND, Negation) benutzt werden.

(c) `getByte(x,n)` Diese Funktion soll das durch `n` angegebene Byte ($0 \leq \mathtt{n} \leq 3$) aus dem Wert `x` extrahieren.

(d) `rotateLeft(x,n)` Die Funktion soll den in Java nicht vorhandenen Rotate-Left Operator für `x` nachbilden. Für das zweite Argument `n` gilt: $0 \leq \mathtt{n} \leq 31$.

Aufgabe 6.2 (Punkte 5+10+5)

Codierung: Für eine Messung soll ein einschrittiger Binärcode entwickelt werden, der 13 Elemente umfasst.

(a) Entwickeln Sie den Code aus einem KV-Diagramm heraus.

(b) Warum kann der Code aus (a) nicht zyklisch-einschrittig sein?

(c) Geben Sie einen zyklisch-einschrittigen Code mit 14 Codewörtern an. Benutzen Sie dazu das rekursive Verfahren aus der Vorlesung.

Aufgabe 6.3 (Punkte 5+15+5+5)

Optimale Codierung: Die folgenden 12 Symbole a_i sind mit ihren Wahrscheinlichkeiten $p(a_i)$ in der Tabelle angegeben:

a_i	a	b	c	d	e	f	g	h	i	j	k	l
$p(a_i)$	0,05	0,1	0,02	0,03	0,1	0,12	0,06	0,12	0,03	0,05	0,3	0,02

(a) Wie groß ist der mittlere Informationsgehalt (die Entropie) H dieser Symbole?

(b) Bilden Sie einen Huffman-Baum und geben sie die zugehörige Symbolcodierung an.

(c) Welche mittlere Codewortlänge H_0 ergibt sich?

(d) Wie groß ist die Redundanz $(H_0 - H)$ ihres Codes?

Aufgabe 6.4 (Punkte 10+10)

2 D-Paritätscode: Wir betrachten den in der Vorlesung vorgestellten zweidimensionalen Paritätscode. Jeweils 49 Datenbits werden als Matrix mit 7×7 Zeilen und Spalten notiert, dann wird zu jeder Zeile und Spalte ein ungerades Paritätsbit hinzugefügt. Ein weiteres Bit ganz unten rechts berechnet sich als Parität der Spalten-Paritätsbits.

$d_{0,0}$	$d_{0,1}$	$d_{0,2}$	$d_{0,3}$	$d_{0,4}$	$d_{0,5}$	$d_{0,6}$	$p_{0,7}$
$d_{1,0}$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	$d_{1,5}$	$d_{1,6}$	$p_{1,7}$
$d_{2,0}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$d_{2,4}$	$d_{2,5}$	$d_{2,6}$	$p_{2,7}$
$d_{3,0}$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	$d_{3,4}$	$d_{3,5}$	$d_{3,6}$	$p_{3,7}$
$d_{4,0}$	$d_{4,1}$	$d_{4,2}$	$d_{4,3}$	$d_{4,4}$	$d_{4,5}$	$d_{4,6}$	$p_{4,7}$
$d_{5,0}$	$d_{5,1}$	$d_{5,2}$	$d_{5,3}$	$d_{5,4}$	$d_{5,5}$	$d_{5,6}$	$p_{5,7}$
$d_{6,0}$	$d_{6,1}$	$d_{6,2}$	$d_{6,3}$	$d_{6,4}$	$d_{6,5}$	$d_{6,6}$	$p_{6,7}$
$p_{7,0}$	$p_{7,1}$	$p_{7,2}$	$p_{7,3}$	$p_{7,4}$	$p_{7,5}$	$p_{7,6}$	$p_{7,7}$

(a) Wie groß ist die Minimaldistanz d dieses Codes? Begründen Sie Ihre Antwort.

(b) Können mit diesem Code alle Einbit-, Zweibit- und Dreibitfehler erkannt und korrigiert werden? Warum?