

Hausaufgaben zum 16. November 2012

Tronje Krabbe 6435002, The-Vinh Jackie Huynh 6388888,
Arne Struck 6326505

15. November 2012

1

(a)

Die Definition des Schwellwertes für die Gleichheit der beiden Zahlen ist nutzerabhängig, das bedeutet bei 2 verschiedenen Nutzern tritt nicht zwangsweise eine Übereinstimmung der Ergebnisse auf. Dies ist fatal, wenn beide Nutzer die selbe Aufgabe mit Verschiedenen Schwellwerten erledigen.

Des weiteren ist nicht klar, in welchem Bereich dieser Wert liegen sollte, dies kann von Aufgabe zu Aufgabe verschieden sein.

Außerdem stellt sich die Frage nach der Größe der beiden Zahlen, die verglichen werden sollen. Es kann in einigen Wertebereichen durch diesen Abgleich trotz großer Wertunterschiede zu angeblicher Gleichheit kommen.

(b)

Man kann die Zahlen durcheinander teilen und, wenn das Ergebnis nahe genug an 1 dran ist, behandelt man sie als "gleich".

Man kann auch das Ergebnis des vorgeschlagenen Vergleichs noch durch eine der beiden Zahlen teilen. Je größer die beiden Zahlen sind, desto näher kommt das Ergebnis größer an die 0 heran und ermöglicht so die Wahl eines noch kleineren Schwellwertes zur möglichen Unterscheidung.

(c)

Das erste Verfahren ist ebenfalls ungenau und geeignete Vergleichswerte nahe 1 zu finden ist ungünstig, da diese in double Kodierung ebenfalls ungenau sind. Das zweite Verfahren eignet sich nur für größere Zahlen, da recht ungenaue Werte für Inputs nahe der 0 mit diesem Verfahren nicht zu erreichen sind. um dies zu beheben ist es möglich die Methode der absoluten Abweichung (für Werte nahe 1 oder 0, je nach Verfahren) für geringe Werte und die der relativen für große Werte zu verwenden.

2**(a)**

```

Die CR LF
SP Lösung CR LF
SP SP der CR LF
SP SP SP Übungsaufgabe CR LF
SP SP SP SP liegt CR LF
SP SP SP SP SP vor CR LF
SP SP SP SP SP SP Ihnen!

```

Wenn man die Steuerzeichen übersetzt erfolgt:

```

Die
Lösung
  der
    Übungsaufgabe
      liegt
        vor
          Ihnen!

```

(b)

Windows, DOS, OS/2, CP/M, TOS (Atari) (an dem Zeilenende CR LF erkennbar)

3**(a)**

ISO-8859-1 benötigt 8 Bit für jedes Zeichen, daraus folgt 800 KByte.
 Die Uni-Code-Codierung nutzt 2 Byte pro Zeichen, also 1600 KByte.
 UTF-8 kann mit einem Byte aber nur die ASCII-Tabelle erreichen, deswegen müssen die Umlaute mit 2 Byte dargestellt werden, daraus folgt:

$$\begin{aligned}
 (0,0056 + 0,0029 + 0,0062 + 0,0031) \cdot 800000 &= 14240 \text{ Sonderzeichen} \\
 \Rightarrow 800000 - 14240 + 2 \cdot 14240 &= 814240 \text{ Byte}
 \end{aligned}$$

(b)

$$\begin{aligned}
 (4DBF - 3400 + 1)_{16} &= 19C0_{16} \\
 (9FCF - 4E00 + 1)_{16} &= 51D0_{16} \\
 19C0_{16} + 51D0_{16} &= 6B90_{16} = 27536_{10} \text{ Symbole}
 \end{aligned}$$

(c)

Bei direkter Unicode-Kodierung müsste der Text $2 \cdot 800000 = 1600000$ Zeichen verwenden, da 2 Byte pro Zeichen verwendet werden. In UTF-8 werden chinesische Zeichen in drei oder vier Bytes kodiert.* Deswegen liegt die genaue Lösung zwischen $3 \cdot 800000 = 2400000$ und $4 \cdot 800000 = 3200000$ Zeichen.

* Siehe: de.wikipedia.org/wiki/Chinesische_Zeichenkodierung#Unicode

4**(a)** $y = 10 \cdot x :$
$$\text{int } y = (x \ll 3) + (x \ll 1)$$
(b) $y = 30 \cdot x :$
$$\text{int } y = (x \ll 5) - (x \ll 1)$$
(c) $y = -48 \cdot x :$
$$\text{int } y = (x \ll 4) - (x \ll 6)$$
(d) $y = 60 \cdot (x + 6) :$
$$a = x + 6$$
$$y = (a \ll 6) - (a \ll 2)$$

5

```
/**
 * RS Hausaufgabe Aufgabe 4.5
 */
public class RSoperationen
{
    /**
     * Aufgabenteil (a)
     */
    public static int bitNor(int x, int y)
    {
        return (~x) & (~y);
    }

    /**
     * Aufgabenteil (b)
     */
    public static int bitXor(int x, int y)
    {
        return ~(x & y) & ~((~x) & (~y));
    }

    /**
     * Aufgabenteil (c)
     * Anmerkung:  $(32 - n) = 32 + (\sim n + 1)$ 
     * (Zweierkomplement zur Darstellung der Subtraktion)
     */
    public static int rotateRight(int x, int n)
    {
        return (x >>> n) | (x << 32 + (~n + 1));
    }

    /**
     * Aufgabenteil (d)
     */
    public static int abs(int x)
    {
        return (x ^ (x >> 31)) + (~(x >> 31) + 1);
    }
}
```

Zu (d):

Um den Betrag einer Zahl darzustellen wird die Information benötigt, ob eine Zahl positiv oder negativ ist. Dazu dient $(x \gg 31)$ dies lässt 2 mögliche Ergebnisse zu 0 (die Zahl ist positiv) oder 0xFFFFFFFF (entspricht einer Kette von Einsen, das bedeutet die Zahl ist negativ).

Wenn x negativ ist, dann wird er durch das XOR $(x \wedge (x \gg 31))$ invertiert und eins addiert, was dem Zweierkomplement entspricht (denn $\sim(x \gg 31)$ entspricht in diesem Fall 0), somit wäre der Betrag einer negativen Zahl ermöglicht.

Wenn x nun positiv ist, dann bliebe x durch das XOR erhalten. Da $(x \gg 31) + 1$ überläuft wird hier 0 addiert (die 1 verschwindet in dem Teil des Überlaufs, der nicht mehr dargestellt wird).

-2^{31} ist nicht mehr als Zweierkomplement (bei int) darstellbar, da es nur einen Wertebereich von -2^{31} bis $2^{31} - 1$ besitzt. Daraus folgt, dass die Methode -2^{31} als Betrag von -2^{31} liefern müsste.