ETH Zürich

Informatik I
Übungsstunde 08

Johannes Odenthal

Hausaufgabenreview · Exercise 08

Funktionsaufrufe zählen

```
1
2 void g(const int n) {
3   if (n == 0) {
4     std::cout << "*";
5     return;
6   }
7   g(n - 1);
8   g(n - 1);
9 }
```

Wie viele Aufrufe hat `g(2)`?

$1 + 2 + 4 = 7$ Aufrufe · 4 Sterne
Insgesamt: $2^{n+1} - 1$ Aufrufe

Hilfsfunktion wiederverwenden

```
1 // Ausschnitt: print_trapezoid
2 // Zeile mit print_row ausgeben,
3 // statt die Schleifen zu kopieren:
4 print_row(base_width,
5           offset_left, height);
```

Palindrome: in der Mitte stoppen

```
1 // Ausschnitt: isPalindromeRecursive
2 if (left >= right) {
3   return true;
4 }
```

Hausaufgabenfragen

Welche Fragen sind bei der letzten Übung offen geblieben?

Binary Representation

Dezimalzahl → Binärzahl

Wie lässt sich 61 als Binärzahl darstellen?

Division durch 2

Division	Rest	
$61 = 2 \times 30 + 1$	1	↑
$30 = 2 \times 15 + 0$	0	|
$15 = 2 \times 7 + 1$	1	|
$7 = 2 \times 3 + 1$	1	|
$3 = 2 \times 1 + 1$	1	|
$1 = 2 \times 0 + 1$	1	|

Reste von unten nach oben

$$61_{10} = 111101_2$$

$$32 + 16 + 8 + 4 + 1 = 61$$

From decimal to binary representation

CodeExpert

Fragen?

Binäre Stellenwerte

Was bedeutet 101.011_2?

Bit	1	0	1	0	1	1
Gewicht	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
Wert	4	0	1	0	$\frac{1}{4}$	$\frac{1}{8}$

$$4 + 1 + \tfrac{1}{4} + \tfrac{1}{8} = 5.375$$

Nachkommastellen bestimmen

Wie entsteht die Binärdarstellung von 0.25?

Rest	× 2	Bit
0.25	0.5	0
0.5	1.0	1

Mit dem Nachkommateil starten

1. Rest mit 2 multiplizieren
2. Ganzzahligen Teil als Bit notieren
3. Nachkommateil als neuen Rest nehmen

Wiederholen, bis der Rest 0 ist

$$0.25 = 0.01_2$$

Rest 0 $\longrightarrow$ fertig

Binary Representation

Berechnet die Binärdarstellungen

1

0.25

2

11.1

Endliche und periodische Darstellung

Endlich

$$0.25 = 0.01_2$$

Periodisch

$$11.1 = 1011.0\overline{0011}_2$$

Der gleiche Rest erzeugt die gleichen nächsten Bits

Fragen?

09:00–09:15

15 Minuten Pause

Normalized FP-systems

Normalisierung im Dezimalsystem

Wie schreiben wir 1230 mit drei Stellen (bzw. signifikanten Ziffern)?

$$1230 = 1.23 \cdot 10^3$$

Basis

10 · Ziffern 0 bis 9

Präzision

3 · die Ziffern 1, 2, 3

Exponent

3 · Faktor 10^3

$1.23 \cdot 10^3 \in F^*(10, 3, -2, 4)$ · Erlaubte Exponenten: –2 bis 4

Normalisierte Darstellung

System: $F^*(b, p, e_{\min}, e_{\max})$

Zahl: $\pm d_0.d_1 ... d_{p-1} \cdot b^e$

Basis b

b verschiedene Ziffern
Dezimal: $b = 10$
$d_i \in \{0, 1, ..., 9\}$
Binär: $b = 2$
$d_i \in \{0, 1\}$

Präzision p

Insgesamt p Ziffern

Exponent e

$e_{\min} \leq e \leq e_{\max}$

$d_0 \neq 0$ · Genau eine von null verschiedene Ziffer vor dem Punkt

Zugehörigkeit zu F*(2,4,–2,2)

Welche Zahlen gehören dazu? Begründet eure Entscheidung

Zahl	Entscheidung und Grund
0	Nein · nicht normalisierbar
$1.000_2 \cdot 2^1$	Ja
$1.001_2 \cdot 2^{-1}$	Ja
$1.0001_2 \cdot 2^{-1}$	Nein · fünf signifikante Ziffern
$1.111_2 \cdot 2^{-2}$	Ja
$1.111_2 \cdot 2^5$	Nein · Exponent ausserhalb –2 bis 2

Some numbers in F*(2,4,–2,2)

Welche Zahlen enthält das System?

Grenzwerte **<- Binär und Dezimal angeben**

– Grösste Zahl
– Kleinste Zahl
– Kleinste nichtnegative Zahl

Anzahl

Wie viele verschiedene Zahlen enthält das System?

Grenzwerte des Systems

Gesucht	Darstellung	Dezimal
Maximum	$1.111_2 \cdot 2^2$	7.5
Minimum	$-1.111_2 \cdot 2^2$	–7.5
Kleinste positive Zahl	$1.000_2 \cdot 2^{-2}$	0.25

Anzahl darstellbarer Zahlen

$$\pm 1.\square\square\square \cdot 2^{e}$$

Vorzeichen

2

Freie Bits

2^3

Produktregel:
3 Bits · je 2 Möglichkeiten

Exponentenbereich

5

–2, –1, 0, 1, 2

$$2 \cdot 2^3 \cdot 5 = 80$$

Computations in F*(2,4,–2,2)

Addiert die beiden Zahlen im System

$$1.001_2 \cdot 2^{-1} + 1.111_2 \cdot 2^{-2}$$

Rechenweg

1. Exponenten angleichen
2. Addieren
3. Normalisieren

Ergebnis

- Auf vier signifikante Ziffern runden
- Mit der exakten Summe vergleichen

Addition und Rundung

1 · Gleicher Exponent

$$
\begin{array}{rl}
 & 1.0010 \times 2^{-1} \\
+ & 0.1111 \times 2^{-1} \\
\hline
 & 10.0001 \times 2^{-1}
\end{array}
$$

2 · Normalisieren

$$1.00001_2 \cdot 2^0$$

3 · Auf p = 4 runden

$$1.000_2 \cdot 2^0 = 1$$

Exakte Summe: 1.03125

Fragen?

Optionales Material

Two’s complement

Welche vier Bits fehlen?

Vier Bits · Übertrag entfällt

```
  0001
+ ????
------
  0000
```

–1

1111

–2

0010 ⟶ 1101 ⟶ 1110
Bits umkehren, dann +1
Wertebereich: –8 bis 7

Ein 10-bit FP-Format

Wie verteilen wir zehn Bits?

Sign	Exponent · 4 Bits	Fraction · 5 Bits
0	1 0 0 0	0 0 1 1 0

$+1.00110_2 \cdot 2^{8-8} = 1.1875$ · Führende 1 wird nicht gespeichert

Bias 8 · Präzision 6 · Exponenten –8 bis 7

Eigenes Modell ohne Sonderwerte · Null ist nicht darstellbar