

Übung zur VL „Grundlagen der Programmierung“

3. Übung

Dr. Zubow

Algorithmen

- Programme = Daten + Algorithmen
- Berechnungsvorschrift (Reihenfolge von Anweisungen, die schrittweise zum Ergebnis führen)
- Eigenschaften:
 - E/A-Spezifikation (Eingabe/Ausgabe)
 - Terminierung (Halteproblem: Endlosschleife)
 - Komplexität

Algorithmen

- Euklidischer Algorithmus (ggT)
 - Bsp. $\text{ggT}(25, 185)$, $\text{ggT}(12, 28)$
 - Eingabe: $a, b \in \mathbb{N}$ mit $ab \neq 0$
 - Ausgabe: $d = \text{ggT}(a, b)$
- Bsp.:
 - $\text{ggT}(25, 185) = 5$, $\text{ggT}(12, 28) = 4$

Euklidischer Algorithmus (ggT)

- klassischer Alg.:

```
EUCLID_OLD(a,b)
  solange b ≠ 0
    wenn a > b
      dann a ← a - b
    sonst b ← b - a
  return a
```
- moderner Alg.:

```
EUCLID_NEW(a,b)
  solange b ≠ 0
    h ← a mod b
    a ← b
    b ← h
  return a
```

Algorithmen

- Aufgabe: Berechnung von π

$$\pi = \lim_{n \rightarrow \infty} \sum_{k=0}^n \left(\frac{1}{16}\right)^k \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right)$$

- Eingabe: ϵ
- Ausgabe: π

Algorithmus - Zahlenraten

- Spieler A sucht sich eine natürliche Zahl z zwischen 1 und 100 aus, die Spieler B erraten soll. Spieler B darf Fragen der Form "Ist z kleiner als m ?" stellen, wobei m eine von B ausgesuchte Zahl ist. A antwortet entweder mit "ja" oder mit "nein".
- Geben Sie einen Algorithmus (in Pseudocode) an, der das Verhalten von Spieler B simuliert. Benutzen Sie die Funktion FRAGE(m), die "wahr" zurückgibt, wenn $z < m$ gilt!
- Wie viele Fragen benötigt Ihr Algorithmus im günstigsten und im ungünstigsten Fall?

Berechnung von π

- Lösung (Pseudocode):

```
READ(eps);
k:=0;
pi :=0.0;
sechzehntel := 1.0;
REPEAT
  summand := 1/(8*k + 1) - 2/(8*k + 4) - 4/(8*k + 5) - 1/(8*k + 6);
  summand := sechzehntel * summand;
  pi := pi + summand;
  sechzehntel := sechzehntel / 16;
  k := k + 1;
UNTIL summand < eps;
WRITE(pi);
WRITE(k);
```

- Komplexität: Anzahl der Operationen "+ -" bzw. "* / DIV MOD" ?

Alphabete & Wörter

- Aufgabe 1
 - Geben Sie eine rekursiv definierte Funktion an, welche das eingegebene Wort in der Mitte spiegelt.
 - Bsp.: Beispiel \rightarrow sieBleip

Graphen

- Was ist ein Graph?
- Geben Sie eine Definition an!
- Was ist eine Adjazenzmatrix?
- Worin liegt der Unterschied zu einem Baum?

Informationsdarstellung

- Was ist ein Bit, Byte?
- Zahlendarstellung
 - Positive Zahlen
 - Negative Zahlen
 - 1bit Vorzeichen, Einerkomplement, Zweierkomplement
 - Problem des Überlaufs

Zahlendarstellung

- Wandeln Sie, sofern möglich, die Dezimalzahlen 13107, -733, 65536 in folgende Darstellung um:
 - Binärzahlen (16 Ziffern),
 - Oktalzahlen (11 Ziffern),
 - Hexadezimalzahlen (8 Ziffern),
 - Zahlen zur Basis 13 (5 Ziffern).
- Verwenden Sie die jeweilige Komplement-darstellung; geben Sie, wenn nötig, führende Nullen mit an!

	13107	-733	65536
Binärzahlen	0011 0011 0011 0011	1111 1101 0010 0011	Zahl nicht darstellbar
Oktalzahlen	00000031463	7777776443	00000200000
Hexadezimalzahlen	00003333	FFFFFFD3	00010000
Zahlen zur Basis 13	05C73	CC888	23AA3

Markov-Algorithmen

- Bsp. für ein formales Maschinenmodell
- Umsetzbar als Interpreter für sogenannte Markov-Tafeln.
- Def.:
 - $A = (a_1, \dots, a_n)$ sei ein Alphabet.
 - A^* sei die Menge der Worte über A .
 - Elementare Einzelschritte sind Regeln der Form $\phi \rightarrow \psi$, wobei $\phi, \psi \in A^*$ (ersetzte Teilwort ϕ durch ψ)
 - Angewandt auf ein Wort $\xi \in A^*$ entsteht somit auf eindeutige Weise ein neues Wort $g[\phi \rightarrow \psi](\xi)$ wie folgt:
 - Ist ϕ ein Teilwort von ξ ($\xi = \mu\phi\nu$, $\mu, \nu \in A^*$) und ist ϕ an dieser Stelle das erste Auftreten (von links) von ϕ in ξ , so ist $g[\phi \rightarrow \psi](\xi) = \mu\psi\nu$
 - Ist ϕ kein Teilwort von ξ , so ist $g[\phi \rightarrow \psi](\xi) = \xi$, d.h. es passiert nichts.

Markov-Algorithmen

- Bsp.: $A=(0, 1)$ und Regel $01 \rightarrow 10$
- $1100101011 \rightarrow 1101001011$
 $\rightarrow 1110001011$
 $\rightarrow 1110010011$
 $\rightarrow 1110100011$
 \rightarrow usw.
- Regel verschiebt also die Nullen nach rechts...
- Markov-Regeln lassen sich in Markov-Tafeln zusammenfassen:
 - 5 Spalten und $0 \dots N$ Zeilen
 - Zeile hat die Form (5-Tupel): $k \phi \psi i j$, wobei $i, j, k \in \mathbb{N}$
 - k ist Zeilennummer, ϕ und ψ stellen die Regel $\phi \rightarrow \psi$ dar, i ist die Nummer der nächsten Zeile (falls Teilwort ϕ gefunden wurde), andernfalls j (Teilwort ϕ nicht gefunden)
 - Ausführung der Markov-Tafeln beginnt in der ersten Zeile (0) und stoppt, sobald zu einer nicht vorhandenen Zeile gegangen wird.

Markov-Algorithmen

- Problem: Markov-Algorithmen arbeiten auf Zeichenketten und nicht auf Zahlen!
- Lösung: Kodierung der Zahlen im unären Zahlensystem.

- Bsp.: "Addiere |"

– $A = \{|\}, X = A^*, Y = A^+ = A^* - \{\epsilon\}$

- Markov-Tafel lautet:

k	ϕ	ψ	i	j
0	ϵ		1	-

- Berechnet: $f(|^n) = |^{n+1}$, $n \in \mathbb{N}$; also die Nachfolgerfkt.

- Weitere: $f(|^n + |^m) = |^{n+m}$; $f(|^n) = |^{2n}$