Übungen zu Programmieren

- Zusatzmaterial Blatt 1 Aufgabe 2 -

Tutoren:

Sven Groppe	groppe@ifis.uni-luebeck.de
Florian Frischat	prog2@ifis.uni-luebeck.de
David Gregorczyk	prog6@ifis.uni-luebeck.de
Dana Linnepe	prog5@ifis.uni-luebeck.de
Sven-Erik Pfeiffer	prog4@ifis.uni-luebeck.de
Markus Weigel	prog1@ifis.uni-luebeck.de
Simon Werner	prog3@ifis.uni-luebeck.de

Institute of Information Systems
University of Lübeck

Programme zur Berechnung von Pi

- Verschiedene Reihen werden zur Berechnung von Pi eingesetzt, z.B.:
  - Li (1949) (Beweis):

$$\frac{\pi}{2} = \sum_{i=0}^{\infty} \frac{i!}{(2 \cdot i + 1)!!} = 1 + \frac{1}{3} + \frac{1 \cdot 2}{3 \cdot 5} + \frac{1 \cdot 2 \cdot 3}{3 \cdot 5 \cdot 7} + \ldots$$

    wobei k!! = 1*3*5*7*…*k für k ungerade
  - Gosper (1974):

$$\pi = 3 + \frac{1}{4 \cdot 5 \cdot 3} \cdot 8 + \frac{1}{4 \cdot 5 \cdot 3} \cdot \frac{2 \cdot 3}{7 \cdot 8 \cdot 3} \cdot 13 + \frac{1}{4 \cdot 5 \cdot 3} \cdot \frac{2 \cdot 3}{7 \cdot 8 \cdot 3} \cdot \frac{3 \cdot 5}{10 \cdot 11 \cdot 3} \cdot 18$$
$$+ \frac{1}{4 \cdot 5 \cdot 3} \cdot \frac{2 \cdot 3}{7 \cdot 8 \cdot 3} \cdot \frac{3 \cdot 5}{10 \cdot 11 \cdot 3} \cdot \frac{4 \cdot 7}{13 \cdot 14 \cdot 3} \cdot 23 + \ldots$$

    Diese Reihe konvergiert schneller als die erste Reihe
  - Bailey, Borwein & Plouffe (1997a):

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

Probleme unserer Lösung

- Unsere Lösung berechnet Pi komplett als Gleitpunktzahl
- Aber: Rundungsfehler bei Verwendung von Gleitpunktzahlen
- => Verwendung von Ganzzahlen und direkte Ausgabe der Dezimalstellen für Pi-Berechnungen

Auszug von Pi-Algorithmen

- Zapfhahn-Algorithmus nach Rabinowitz & Wagon (1995)
  - Verwendung der Reihe nach Li (1949)
  - Abwandlung
    - Vermutlich kürzestes C-Programm (158 Zeichen) zur Berechnung von Pi:

```
int a=10000,b,c=8400,d,e,f[8401],g;
main(){for(;b-c;)f[b++]=a/5;
for(;d=0,g=c*2;c-=14,printf("%.4d",e+d/a),e=d%a)
for(b=c;d+=f[b]*a,f[b]=d%--g,d/=g--,--b;d*=b);}
```

    - Nicht ganz korrekt, da eine Überlaufbehandlung fehlt
    - Aber: Korrekte Berechnung der ersten 2400 Dezimalstellen
    - Kolmogorow-Komplexität
      - Länge des kürzesten Programms, das ein gegebenes Problem löst

- Bailey, Borwein & Plouffe (1997b)
  - Direkte Berechnung einer Hexadezimalziffer ohne vorherige Ziffern zu berechnen
    - Verwendung der Reihe nach Bailey, Borwein & Plouffe (1997a)

Literatur

Jörg Arndt & Christoph Haenel (2000). *Pi — Algorithmen, Computer, Arithmetik*. Springer-Verlag, Berlin, Heidelberg, New York, 2nd edition. ISBN 3-540-66258-8.

R. W. Gosper (1974). Acceleration of series. *Memo M. I. T. Artificial Intelligence Laboratory* **304**.

Donald E. Knuth (1981). *The Art of Computer Programming, vol.2, Seminumerical Algorithms*. Addison-Wesley, Reading MA, 2nd edition.

J.C.R Li (1949). Problem E854. *American Mathematical Monthly* **56**, 633–635.

S. D. Rabinowitz & S. Wagon (1995). A Spigot Algorithm for Pi. *American Mathematical Monthly* **103**, 195–203.

David H. Bailey, Jonathan M. Borwein, Peter B. Borwein & S. Plouffe (1997a). The quest for pi. *The Mathematical Intelligencer* **19**(1), 50–57.

David H. Bailey, Peter B. Borwein & Simon Plouffe (1997b). On the rapid computation of various polylogarithmic constants. *Mathematics of Computation* **66**(218), 903–913. URL `http://www.cecm.sfu.ca/~pborwein/`.