

Title: Seidl: GAD (13.04.2016)

Date: Wed Apr 13 13:21:50 CEST 2016

Duration: 43:48 min

Pages: 18

Kreis zeichnen: mit Multiplikation

Algorithmus Bresenham1: zeichnet Kreis mit Radius R

```

x = 0; y = R;
plot(0, R); plot(R, 0); plot(0, -R); plot(-R, 0);
F =  $\frac{5}{4} - R$ ;
while x < y do
  if F < 0 then
    F = F + 2 * x + 1;
  else
    F = F + 2 * x - 2 * y + 2;
    y = y - 1;
  x = x + 1;
  plot(x, y); plot(-x, y); plot(-y, x); plot(-y, -x);
  plot(y, x); plot(y, -x); plot(x, -y); plot(-x, -y);

```

Es geht sogar noch etwas schneller!

Kreis zeichnen: mit Addition / Subtraktion

- Ersetzung der Korrekturterme für F :

$$\begin{aligned}
 F = F + 2x + 1 &\rightarrow F = F + d_E \\
 F = F + 2x - 2y + 2 &\rightarrow F = F + d_{SE}
 \end{aligned}$$

mit $d_E = 2x + 1$ und $d_{SE} = 2x - 2y + 2$

- Anfangswerte:

$$\begin{aligned}
 d_E(0, R) &= 2 \cdot 0 + 1 = 1 \\
 d_{SE}(0, R) &= 2 \cdot 0 - 2 \cdot R + 2 = 2 - 2 \cdot R
 \end{aligned}$$

- Updates nach rechts (E) und nach unten rechts (SE):

$$\begin{aligned}
 d_E(x + 1, y) &= 2 \cdot (x + 1) + 1 = d_E(x, y) + 2 \\
 d_{SE}(x + 1, y) &= 2 \cdot (x + 1) - 2 \cdot y + 2 = d_{SE}(x, y) + 2 \\
 d_E(x + 1, y - 1) &= 2 \cdot (x + 1) + 1 = d_E(x, y) + 2 \\
 d_{SE}(x + 1, y - 1) &= 2 \cdot (x + 1) - 2 \cdot (y - 1) + 2 = d_{SE}(x, y) + 4
 \end{aligned}$$

Kreis zeichnen: mit Addition / Subtraktion

- Der Bruch $\frac{5}{4}$ kann durch 1 ersetzt werden, weil sich F immer um eine ganze Zahl ändert.

- D.h.

$$F = \frac{5}{4} - R + k < 0$$

ist äquivalent zu

$$F = 1 - R + k < 0$$

- Vorteil:

nur noch ganze Zahlen!

Kreis zeichnen: mit Addition / Subtraktion

Algorithmus Bresenham2: zeichnet Kreis mit Radius R

$x = 0$; $y = R$; $\text{plot}(0, R)$; $\text{plot}(R, 0)$; $\text{plot}(0, -R)$; $\text{plot}(-R, 0)$;

$F = 1 - R$; $d_E = 1$; $d_{SE} = 2 - R - R$;

while $x < y$ **do**

if $F < 0$ **then**

$F = F + d_E$;

$d_{SE} = d_{SE} + 2$;

else

$F = F + d_{SE}$;

$y = y - 1$;

$d_{SE} = d_{SE} + 4$;

$x = x + 1$; $d_E = d_E + 2$;

$\text{plot}(x, y)$; $\text{plot}(-x, y)$; $\text{plot}(-y, x)$; $\text{plot}(-y, -x)$;

$\text{plot}(y, x)$; $\text{plot}(y, -x)$; $\text{plot}(x, -y)$; $\text{plot}(-x, -y)$;

Bresenham-Algorithmus

- Ab Anfang der 1960er Jahre hat JACK BRESENHAM Algorithmen zur Linien- und Kreisdarstellung entwickelt.
- Diese verwenden nur einfache Additionen ganzer Zahlen.
- Sie sind damit deutlich schneller als die naiven Ansätze.

Multiplikation langer Zahlen

Schulmethode:

- gegeben Zahlen a und b
- multipliziere a mit jeder Ziffer von b
- addiere die Teilprodukte

$$\begin{array}{r}
 5\ 6\ 7\ 8 \cdot \boxed{4}\ 3\ 2\ 1 \\
 \hline
 2\ 2\ 7\ 1\ 2 \\
 1\ 7\ 0\ 3\ 4 \\
 1\ 1\ 3\ 5\ 6 \\
 5\ 6\ 7\ 8 \\
 \hline
 2\ 4\ 5\ 3\ 4\ 6\ 3\ 8
 \end{array}$$

Multiplikation langer Zahlen

Schulmethode:

- gegeben Zahlen a und b
- multipliziere a mit jeder Ziffer von b
- addiere die Teilprodukte

$$\begin{array}{r}
 5\ 6\ 7\ 8 \cdot 4\ 3\ 2\ 1 \\
 \hline
 2\ 2\ 7\ 1\ 2 \\
 1\ 7\ 0\ 3\ 4 \\
 1\ 1\ 3\ 5\ 6 \\
 5\ 6\ 7\ 8 \\
 \hline
 2\ 4\ 5\ 3\ 4\ 6\ 3\ 8
 \end{array}$$

Aufwand

- Wenn die Zahlen klein sind, ist der Aufwand ok.
 - Aber wenn die Zahlen sehr lang sind, kann man das Produkt dann schneller ausrechnen als mit der Schulmethode?
- ⇒ Wie wollen wir die Zeit oder den Aufwand überhaupt messen?
- Am besten nicht in Sekunden, die irgendein Rechner braucht, denn das könnte für einen anderen Rechner eine ganz andere Zahl sein.
 - Außerdem werden die Computer ja von Generation zu Generation immer schneller und leistungsfähiger.
- ⇒ Wir zählen **Grundoperationen**: Operationen, die man in einem einzigen Schritt bzw. in einer konstanten Zeiteinheit ausführen kann.

Multiplikation langer Zahlen

Schulmethode:

- gegeben Zahlen a und b
- multipliziere a mit jeder Ziffer von b
- addiere die Teilprodukte

$$\begin{array}{r}
 5\ 6\ 7\ 8 \cdot 4\ 3\ 2\ 1 \\
 \underline{2\ 2\ 7\ 1\ 2} \\
 \quad 1\ 7\ 0\ 3\ 4 \\
 \quad \quad 1\ 1\ 3\ 5\ 6 \\
 \quad \quad \quad 5\ 6\ 7\ 8 \\
 \hline
 2\ 4\ 5\ 3\ 4\ 6\ 3\ 8
 \end{array}$$

Aufwand

- Wenn die Zahlen klein sind, ist der Aufwand ok.
 - Aber wenn die Zahlen sehr lang sind, kann man das Produkt dann schneller ausrechnen als mit der Schulmethode?
- ⇒ Wie wollen wir die Zeit oder den Aufwand überhaupt messen?
- Am besten nicht in Sekunden, die irgendein Rechner braucht, denn das könnte für einen anderen Rechner eine ganz andere Zahl sein.
 - Außerdem werden die Computer ja von Generation zu Generation immer schneller und leistungsfähiger.
- ⇒ Wir zählen **Grundoperationen**: Operationen, die man in einem einzigen Schritt bzw. in einer konstanten Zeiteinheit ausführen kann.

Grundoperation

- Multiplikation von zwei Ziffern: $x \cdot y = ?$
Das Ergebnis besteht aus (höchstens) zwei Ziffern u (Zehnerstelle) und v (Einerstelle), also

$$x \cdot y = 10 \cdot u + v$$

- Addition von drei Ziffern: $x + y + z = ?$
Auch hier besteht das Ergebnis aus (höchstens) zwei Ziffern u (Zehnerstelle) und v (Einerstelle), also

$$x + y + z = 10 \cdot u + v$$

Wir benutzen hier drei Ziffern als Summanden, weil wir später Überträge berücksichtigen wollen.

Analyse der Addition

- Zahl plus Zahl:

$$\begin{array}{r}
 6917 \\
 4269 \\
 \hline
 1101 \text{ (Carry)} \\
 \hline
 11186
 \end{array}$$

Zur Addition zweier Zahlen mit jeweils n Ziffern brauchen wir n Additionen von 3 Ziffern, also n Grundoperationen.

Ergebnis: Zahl mit $n + 1$ Ziffern

Analyse des Teilprodukts

- Zahl mal Ziffer:

$$\begin{array}{r}
 5678 \cdot 4 \\
 \hline
 2312 \\
 28 \\
 24 \\
 20 \\
 \hline
 22712
 \end{array}$$

Zur Multiplikation einer Zahl bestehend aus n Ziffern mit einer einzelnen Ziffer brauchen wir

- ▶ n Multiplikationen von 2 Ziffern und
- ▶ $n + 1$ Additionen von 3 Ziffern, wobei in der letzten Spalte eigentlich nichts addiert werden muss,

also $2n[+1]$ Grundoperationen.

Ergebnis: Zahl mit $n + 1$ Ziffern

Analyse des Produkts

- Zahl mal Zahl:

$$\begin{array}{r}
 5678 \cdot 4321 \\
 \hline
 22712000 \\
 1703400 \\
 113560 \\
 5678 \\
 \hline
 24534638
 \end{array}$$

Zur Multiplikation zweier Zahlen mit jeweils n Ziffern brauchen wir

- ▶ n Multiplikationen einer n -Ziffern-Zahl mit einer Ziffer, also $n \cdot (2n[+1]) = 2n^2[+n]$ Grundoperationen
- ▶ Zwischenergebnisse sind nicht länger als das Endergebnis ($2n$ Ziffern), also $n - 1$ Summen von Zahlen mit $2n$ Ziffern, also $(n - 1) \cdot 2n = 2n^2 - 2n$ Grundoperationen

Insgesamt: $4n^2 - [2]n$ Grundoperationen

Analyse des Produkts

- Zahl mal Zahl:

$$\begin{array}{r}
 5678 \cdot 4321 \\
 \hline
 22712 \\
 17034 \\
 11356 \\
 5678 \\
 \hline
 24534638
 \end{array}$$

Genauer:

- ▶ Beim Aufsummieren der Zwischenergebnisse muss man eigentlich jeweils nur Zahlen bestehend aus $n + 1$ Ziffern addieren. Das ergibt $(n - 1)(n + 1) = n^2 - 1$ Grundoperationen.

Insgesamt hätte man damit $3n^2[+n] - 1$ Grundoperationen.

Geht es besser?

Frage:

5 n log n

- Ist das überhaupt gut?
- Vielleicht geht es ja schneller?
- Was wäre denn überhaupt eine signifikante Verbesserung?
- Vielleicht irgendetwas mit $2n^2$?
- Das würde die Zeit auf ca. $2/3$ des ursprünglichen Werts senken.
- Aber bei einer Verdoppelung der Zahlenlänge hätte man immer noch eine Vervielfachung der Laufzeit.
- Wir werden diese Frage später beantworten ...

Algorithmen-Beispiele

- Rolf Klein und Tom Kamphans:
Der Pledge-Algorithmus: Wie man im Dunkeln aus einem Labyrinth entkommt
- Dominik Sibbing und Leif Kobbelt:
Kreise zeichnen mit Turbo
- Arno Eigenwillig und Kurt Mehlhorn:
Multiplikation langer Zahlen (schneller als in der Schule)
- Diese und weitere Beispiele:



Taschenbuch der Algorithmen (Springer, 2008)

Effizienzmessung

Ziel:

- Beschreibung der Performance von Algorithmen
- möglichst genau, aber in kurzer und einfacher Form

Exakte Spezifikation der Laufzeit eines Algorithmus
(bzw. einer DS-Operation):

- Menge I der Instanzen
- Laufzeit des Algorithmus $T: I \mapsto \mathbb{N}$

Problem: T sehr schwer exakt bestimmbar bzw. beschreibbar

Lösung: **Gruppierung** der Instanzen (meist nach **Größe**)