Script generated by TTT

Title: Seidl: GAD (13.04.2016)

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

Duration: 43:48 min

Pages: 18

Einführund

Beispiele

Kreis zeichnen: mit Addition/Subtraktion

• Ersetzung der Korrekturterme für *F*:

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

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

Anfangswerte:

$$d_E(0,R) = 2 \cdot 0 + 1 = 1$$

 $d_{SE}(0,R) = 2 \cdot 0 - 2 \cdot R + 2 = 2 - 2 \cdot R$

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

$$d_{E}(x+1,y) = 2 \cdot (x+1) + 1 = d_{E}(x,y) + 2$$

$$d_{SE}(x+1,y) = 2 \cdot (x+1) - 2 \cdot y + 2 = d_{SE}(x,y) + 2$$

$$d_{E}(x+1,y-1) = 2 \cdot (x+1) + 1 = d_{E}(x,y) + 2$$

$$d_{SE}(x+1,y-1) = 2 \cdot (x+1) - 2 \cdot (y-1) + 2 = d_{SE}(x,y) + 4$$

Einführung

Beispie

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;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F = F + 2 * x + 1;

F =
```

Es geht sogar noch etwas schneller!

H. Seidl (TUM

GAE

16 30

Finführung

plot(y, x); plot(y, -x); plot(x, -y); plot(-x, -y);

Kreis zeichnen: mit Addition/Subtraktion

- Der Bruch 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; \quad y = R; \quad \text{plot}(0, R); \, \text{plot}(0, -R); \, \text{plot}(-R, 0);
F = 1 - R; \quad d_E = 1; \quad 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; \quad d_E = d_E + 2;
\text{plot}(x, y); \quad \text{plot}(-x, y); \quad \text{plot}(-y, x); \quad \text{plot}(-y, -x);
\text{plot}(y, x); \quad \text{plot}(y, -x); \quad \text{plot}(-x, -y);

H. Seid (TUM)

GAD

SS16

33
```

Bresenham-Algorithmus

Ab Anfang der 1960er Jahre hat Jack Bresenham
Algorithmen zur Linien- und Kreisdarstellung entwickelt.

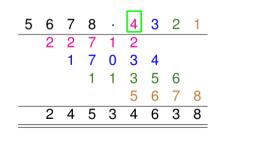
Diese verwenden nur einfache Additionen ganzer Zahlen.

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



Multiplikation langer Zahlen

Schulmethode:

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

5	6	1	8	•	4	3	2	1
	2	2	7	1	2			
		1	7	0	3	4		
			1	1	3	5	6	
					5	6	7	8
	2	4	5	3	4	6	3	8

◄□▶
□▶
4□▶
4□▶
4□▶
4□▶
4□▶
4□▶
4□▶
4□▶
4□▶

H. Seidl (TUM)

GAD

SS'16



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.

Einführung

eispiele

• 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.

Einführung

Beispie

Multiplikation langer Zahlen

Schulmethode:

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

5	6	7	8		4	3	2	1
	2	2	7	1	2			
		1	7	0	3	4		
			1	1	3	5	6	
					5	6	7	8
	2	4	5	3	4	6	3	8

L Seidl (TUM)

GAD

SS'16 3

Beispiele

Grundoperation

Multiplikation von zwei Ziffern: x · 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.



Zahl plus Zahl:

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

Ergebnis: Zahl mit n + 1 Ziffern

<ロ> < □> < □> < □> < 豆> < 豆> < 豆> < ⊙< ○</p>

H. Seidl (TUM)

GAD

SS'16 38

Finführu

Beispie

Analyse des Teilprodukts

Zahl mal Ziffer:

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,

Beispiele

also 2n[+1] Grundoperationen.

Ergebnis: Zahl mit n + 1 Ziffern

Finführung

ieniala

Analyse des Produkts

Zahl mal Zahl:

5	6	7	8		4	3	2	1
	2	2	7	1	2	0	0	0
		1	7	0	3	4	0	0
			1	1	3	5	6	0
					5	6	7	8
	2	4	5	3	4	6	3	8

Zur Multiplikation zweier Zahlen mit jeweils n Ziffern brauchen wir

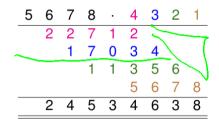
- ► *n* Multiplikationen einer *n*-Ziffern-Zahl mit einer Ziffer, also $n \cdot (2n[+1]) = 2n^2[+n]$ Grundoperationen
- Zwischenergebnisse sind nicht länger als das Endergebnis (2n Ziffern), also n-1 Summen von Zahlen mit 2n Ziffern, also $(n-1)\cdot 2n=2n^2-2n$ Grundoperationen

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

D ... 1. 1. . .

Analyse des Produkts

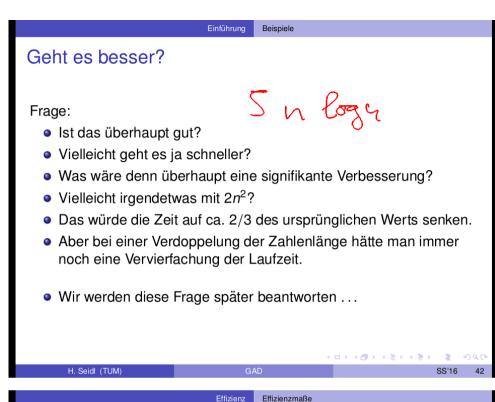
Zahl mal Zahl:



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.



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 → IN

Problem: T sehr schwer exakt bestimmbar bzw. beschreibbar

Lösung: Gruppierung der Instanzen (meist nach Größe)

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)