Algorithmus: vollständige Beschreibung einer Abfolge elementarer Operationen, meist zur Lösung eines Problems

im Alltag:
- Kochrezept
- Bauplan für LEGO, IKEA, ...
- schriftliche Multiplikation

Welche Probleme werden gelöst? Welche Operationen werden gelöst?
→ Mit einfachen Mitteln viel erreichen

in der Informatik: Computer können beliebige Algorithmen auf Daten schnell ausführen

$\approx 10^9$ Operationen/s

Informatik wird oft als Wissenschaft der Algorithmen definiert; Computer nur Werkzeug

in dieser Vorlesung:
- Entwurf und Analyse von Algorithmen
- algorithmische Denkweise

heute: erste Beispiele

Denkweise wichtiger als konkrete Algorithmen

Schriftliche Multiplikation / Schulalgorithmus

$$\begin{array}{rrcrr} a_1 & a_0 & & b_1 & b_0 \\ 8 & 7 & \cdot & 4 & 3 \end{array}$$

$$\begin{array}{r l} 21 & a_0 \cdot b_0 \\ 24 & a_1 \cdot b_0 \\ 28 & a_0 \cdot b_1 \\ 32 & a_1 \cdot b_1 \\ \hline 3741 & \end{array}$$

berechne 4 Teilprodukte

summiere

Korrektheit:

$$(10 \cdot a_1 + a_0) \cdot (10 \cdot b_1 + b_0)$$

Algorithmus gibt richtiges Ergebnis aus

$$= a_0 b_0 + 10 \cdot (a_1 b_0 + a_0 b_1) + 100 \cdot a_1 b_1 \quad \checkmark$$

n-stellige Zahlen:

berechne alle Teilprodukte

$$\underbrace{a_0 b_0, \ldots, a_{n-1} b_0}_{n}, \underbrace{a_0 b_1, \ldots, a_{n-1} b_1}_{n}, \ldots, \underbrace{a_0 b_{n-1}, \ldots, a_{n-1} b_{n-1}}_{n}$$

$\rightsquigarrow$ n^2 Multiplikationen 1-stelliger Zahlen

geht es besser? weniger 1-stellige Multiplikationen?

Ja! (Anatoly Karatsuba, 1960) Russischer Mathematiker

Kolmogorov vermutete n^2 ist best möglich

Karatsuba besuchte als Student das Seminar von Kolmogorov

Karatsuba's Algorithmus

Fall $n=2$

$a_1\, a_0 \quad b_1\, b_0$

$87 \cdot 43$

21	$a_0 b_0$
32	$a_1 b_1$
53	$a_0 b_0 + a_1 b_1$
-1	$-(a_1 - a_0)(b_1 - b_0)$
3741	

3 Multiplikationen 1-stelliger Zahlen

$\rightsquigarrow$ weniger als Schulalgorithmus

Korrektheit

$$a_0 b_0 + 100 \cdot a_1 b_1 + 10 \cdot (\underbrace{a_0 b_0 + a_1 b_1 - (a_1 - a_0)(b_1 - b_0)}_{a_0 b_1 + a_1 b_0})$$

$$= (10 \cdot a_1 + a_0) \cdot (10 \cdot b_1 + b_0) \quad \checkmark$$

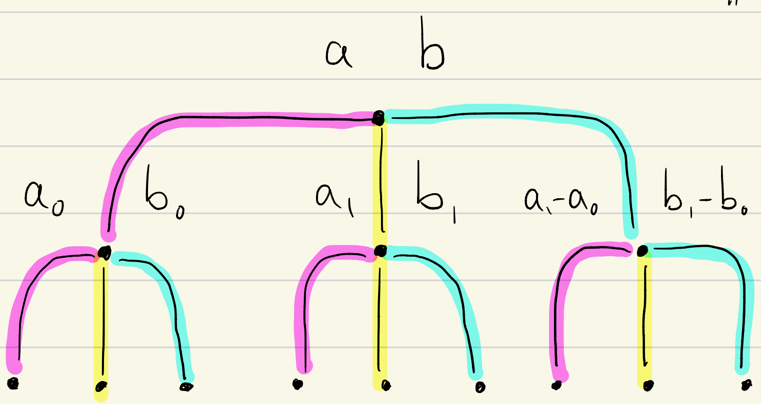
Fall $n=4$

$$\underbrace{8\,7}_{a_1}\;\underbrace{6\,5}_{a_0} \cdot \underbrace{4\,3}_{b_1}\;\underbrace{2\,1}_{b_0}$$

1365	$a_0 b_0$
3741	$a_1 b_1$
4622	$\underbrace{a_0 b_0 + a_1 b_1 - (a_1 - a_0)\cdot(b_1 - b_0)}$
37873565	3 Multiplikationen 2-stelliger Zahlen

Wie weiter?

Bildlich



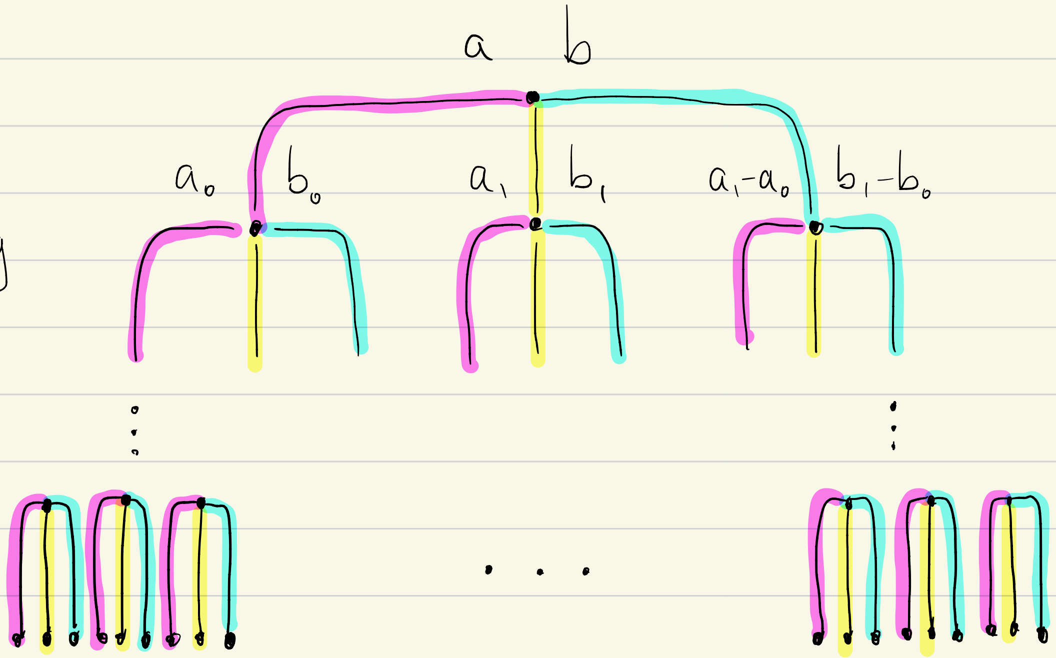
	# Multiplikationen
4-stellig	1
2-stellig	3
1-stellig	9

Rekursion: führe Lösung zurück auf Lösungen kleinerer Eingaben

"Divide-and-Conquer" zerlege Eingaben in kleinere

Fall $n = 2^k$, $k \in \mathbb{N}$ beliebig

	# Multiplikationen
2^k-stellig	1
2^{k-1}-stellig	3
$\vdots$	$\vdots$
2-stellig	3^{k-1}
1-stellig	3^k



(statt $n^2 = 4^k$)

$\underline{k=10:}$ mehr als 10-mal "billiger" als Schulalgorithmus

$\underline{k=20:}$ —— " —— 100-mal —— " ——

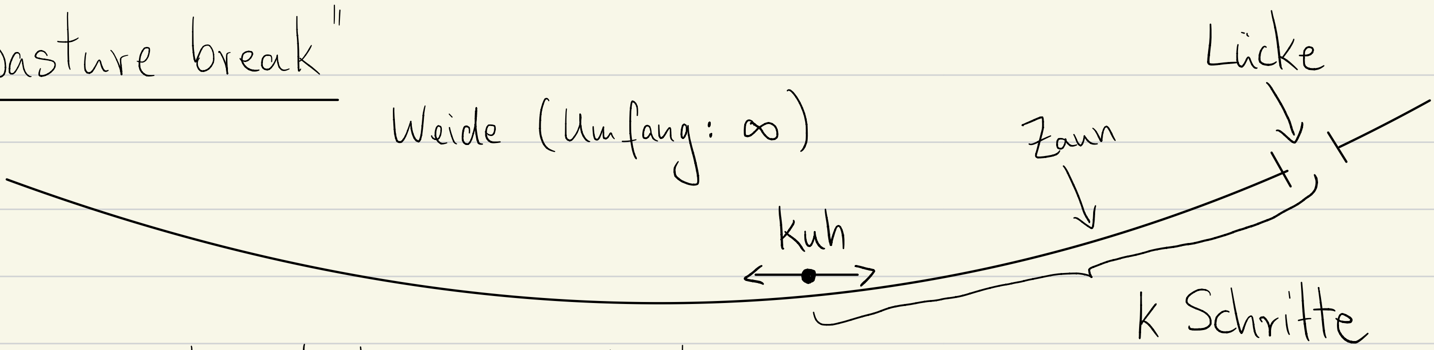
$\vdots$ $\vdots$

geht es besser?

Ja! $\underbrace{C \cdot 2^k \cdot k}_{\text{viel komplizierterer Algorithmus}}$ für grosse Konstante C (Harvey, v.d. Hoefen, 2019)

In Vorlesung geht es nicht darum besser multiplizieren zu lernen, sondern darum naheliegende Lösungsverfahren zu hinterfragen und dramatisch zu verbessern

"pasture break"



Kurzsichtige Kuh sucht Lücke im Zaun

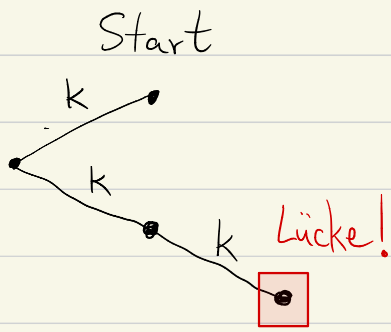
Ziel: möglichst wenig Schritte

Algorithmus 0: k Schritte links, 2k Schritte rechts

Analyse

Fall "Lücke links": k Schritte

Fall "Lücke rechts": 3k Schritte



worst case: 3k Schritte

was wenn k unbekannt?

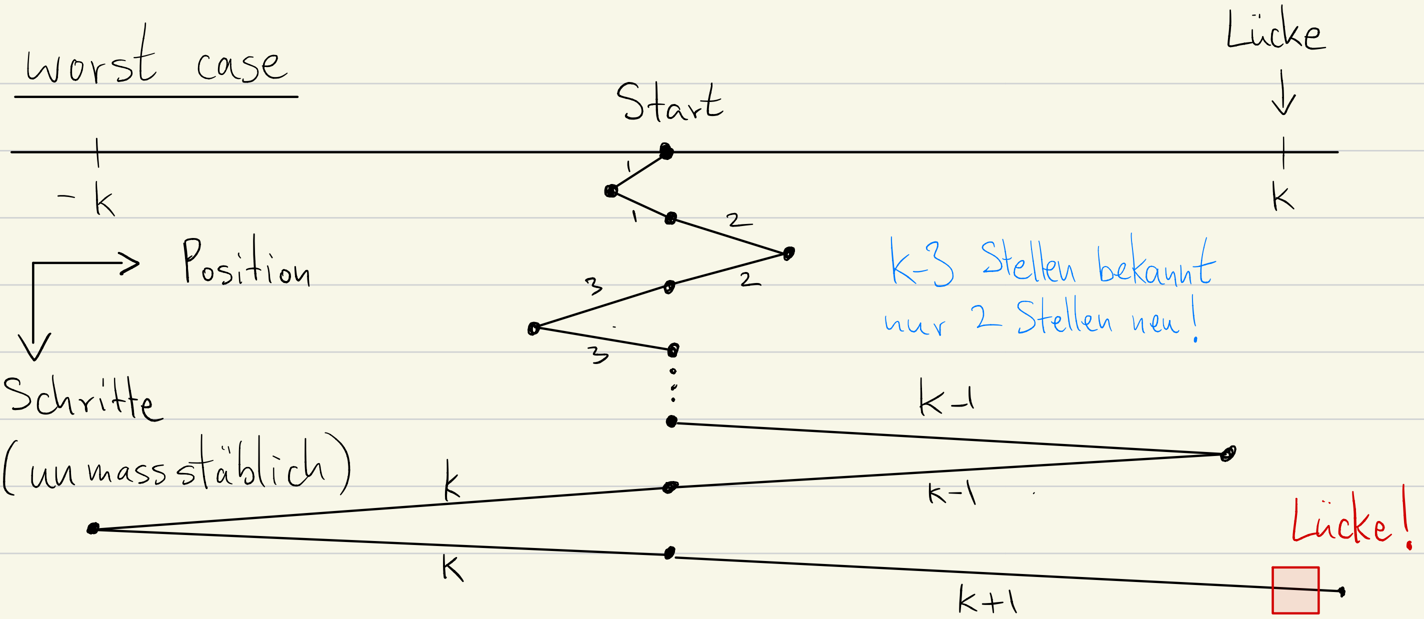
Algorithmus 1: 1 links, zurück zum Start,

2 rechts, ——— " ———

3 links, ——— " ———

⋮

worst case



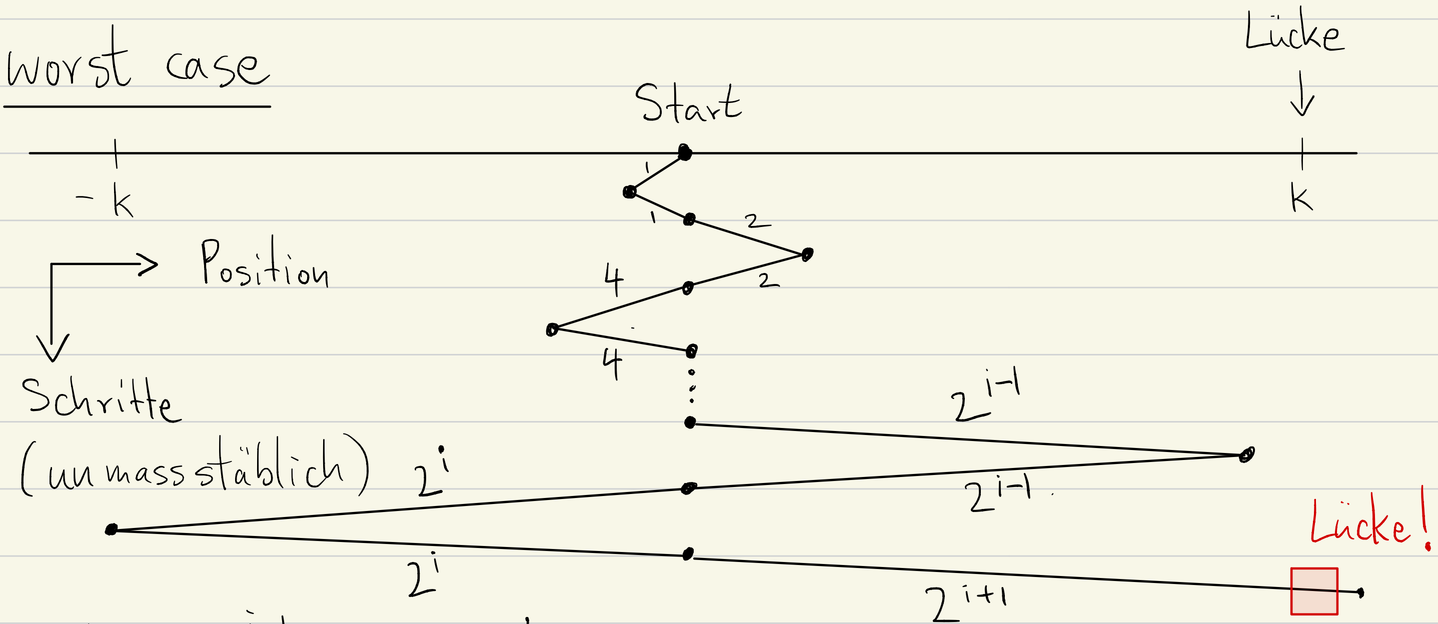
Analyse: $\underbrace{2\cdot 1 + 2\cdot 2 + \dots + 2\cdot(k-1) + 2\cdot k}_{} + k$

Übungsblatt $\boxed{=}$ $k\cdot(k+1) \quad + k = k\cdot(k+2)$

geht es besser? Ja!

Algorithmus 2: verdoppeln statt nur um 1 erhöhen

worst case



wobei $2^{i-1} < k \leq 2^i$, $i \in \mathbb{N}$

Analyse: $2\cdot(1+2+4+\dots+2^{i-1}+2^i)+k$

$\boxed{=}$ $2\cdot\underbrace{(2^{i+1}-1)}_{<4\cdot 2^{i-1}<4\cdot k} + k < 9\cdot k$

später heute

geht es besser?

nicht im worst case!

Beweisprinzip: Induktion

Summe der ersten k 2-er Potenzen: $S_k = 1+2+4+8+\dots+2^{k-1}$

also: $S_1 = 1$, $S_2 = 1+2$, $S_3 = 1+2+4$, $S_4 = 1+2+4+8$, ...

Behauptung: für alle $k \in \{1,2,\dots\} = \mathbb{N}$ gilt $S_k = 2^k - 1$

Induktionsanfang $k=1$:

$$S_1 = 1 = 2^1 - 1 \quad \checkmark$$

Induktionsschritt $k \to k+1$:

$$S_{k+1} = 1+2+4+8+\dots+2^{k-1}+2^k$$

$$= S_k + 2^k \quad \text{(per Definition von } S_{k+1} \text{ und } S_k\text{)}$$

$$= 2^k - 1 + 2^k \quad \text{(Induktionshypothese)}$$

$$= 2^{k+1} - 1 \quad \checkmark$$

Allgemeine Form

Aussage $A(k)$ z.B. "es gilt $1+2+4+8+\dots+2^{k-1}=2^k-1$"

zu beweisen: für alle $k\in\mathbb{N}$ gilt $A(k)$

Induktionsbeweis

Ind. anfang (I.A.): zeige $A(1)$

Ind. schritt (I.S.): zeige für alle $k\in\mathbb{N}$:

wenn $A(k)$ gilt, dann gilt auch $A(k+1)$

d.h.: wir müssen $A(k+1)$ beweisen

aber dürfen dabei $A(k)$ annehmen

(Ind. hypothese, I.H.)

k	1	2	3	4	...
$A(k)$	✓	✓	✓	✓	...

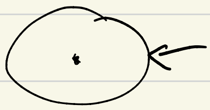
I.A. I.S. I.S. I.S.

$k=1$ $k=2$ $k=3$

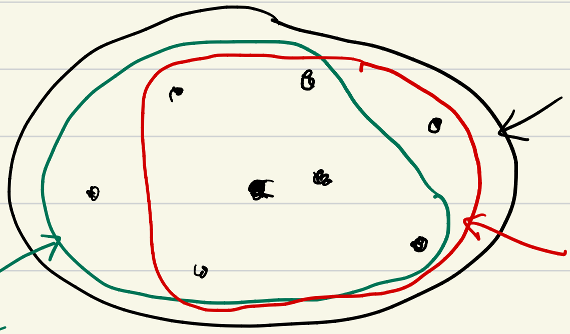
Einschub: Tücken bei Induktionsbeweise

Aussage $A(k)$: in jeder Herde von k Kühen haben alle diesselbe Farbe

beweise per Induktion: $A(k)$ gilt für alle $k \in \mathbb{N}$, $k \geq 1$?

$k=1$:  1 Kuh ✓

$k \to k+1$:



k+1 Kühe

k Kühe ⇝ gleichfarbig (I.H.)

k Kühe ⇝ gleichfarbig (I.H.)

beide k-Herden teilen sich Kühe

⇝ alle k+1 Kühe gleiche Farbe

Fehler: "$k \to k+1$" Beweis falsch für $k=1$

(weil beide k-Herden disjunkt)

Star Suche

Problem: finde "Star" unter n Personen

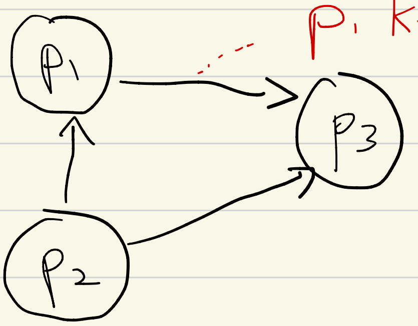
$p_1, \dots, p_n \qquad (n \geq 2)$

Definition: p_s ist Star falls

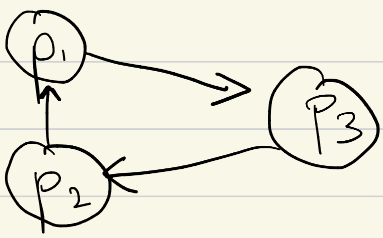
- jede andere Person kennt p_s
- p_s kennt keine andere Person

Beispiele:

"p_1 kennt p_3"



Star: p_3



kein Star

immer ≤ 1 Star

elementare Operation: frage $p_i \xrightarrow{?} p_j$

naiver Algorithmus: alle $n \cdot (n-1)$ Fragen

geht es besser? (nächste Woche)