

1 Einführung: Algorithmen

*Algorithmen und Datenstrukturen
WS 2020/21*

Prof. Dr. Sándor Fekete

Konzentration



Konzentration



Konzentration



Konzentration



Psychological SCIENCE


A Journal of the
Association for
Psychological Science

Home

OnlineFirst

All Issues

Subscribe

RSS 

Email Alerts

The Pen Is Mightier Than the Keyboard Advantages of Longhand Over Laptop Note Taking

Pam A. Mueller¹

Daniel M. Oppenheimer²

¹Princeton University

²University of California, Los Angeles

Pam A. Mueller, Princeton University, Psychology Department, Princeton, NJ 08544

E-mail: pamuele@princeton.edu

Author Contributions Both authors developed the study concept and design. Data collection was supervised by both authors. P. A. Mueller analyzed the data under the supervision of D. M. Oppenheimer. P. A. Mueller drafted the manuscript, and D. M. Oppenheimer revised the manuscript. Both authors approved the final version for submission.



Abstract

Taking notes on laptops rather than in longhand is increasingly common. Many researchers have suggested that laptop note taking is less effective than longhand note taking for learning. Prior studies have primarily focused on students' capacity for multitasking and distraction when using laptops. The present research suggests that even when laptops are used solely to take notes, they may still be impairing learning because their use results in shallower processing. In three studies, we found that

Konzentration



Konzentration



Konzentration

Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Skript

Algorithmen und Datenstrukturen

Konzentration

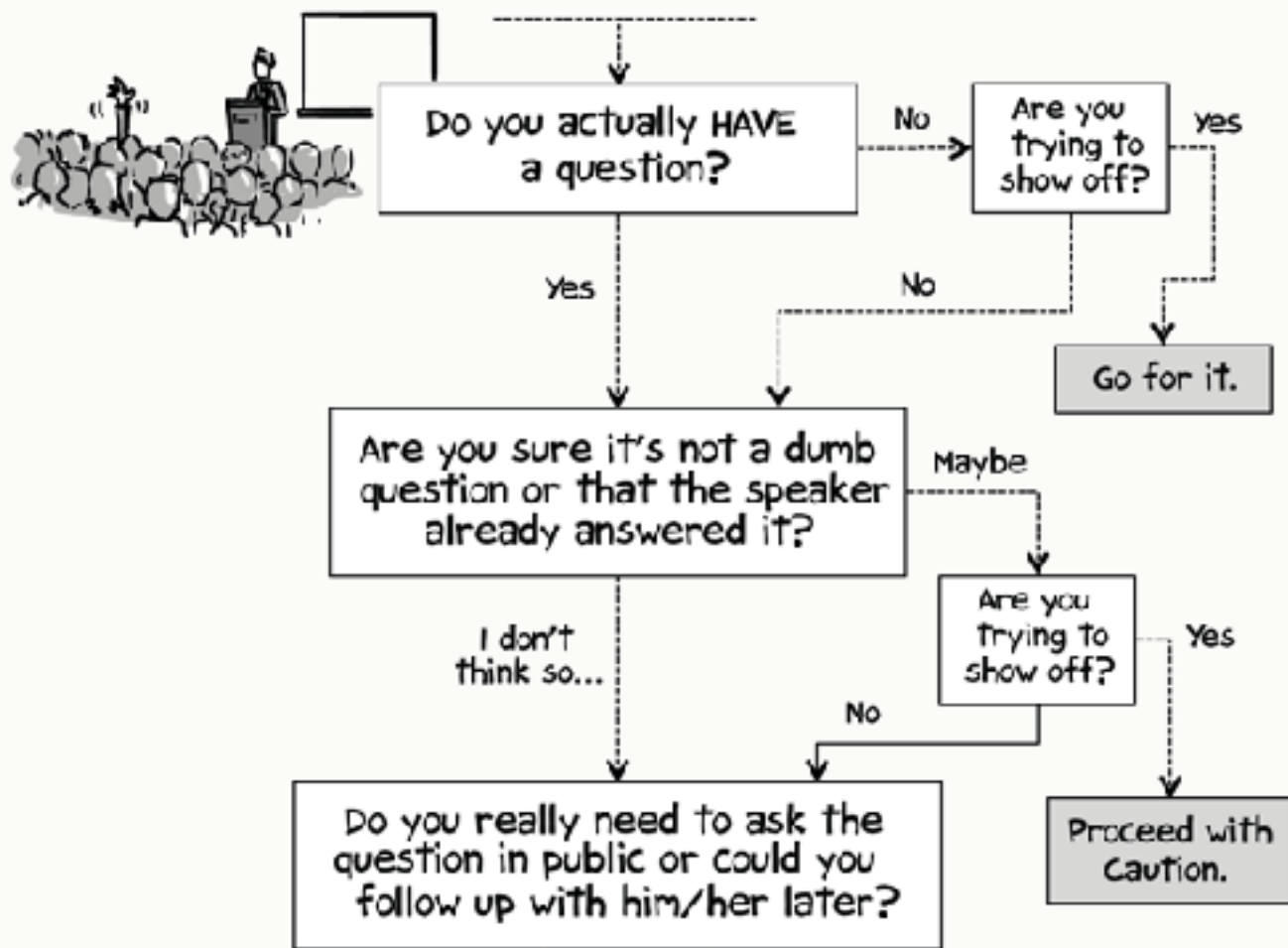


Fragen?



Fragen!

Should you ask a Question during Class?



Doesn't matter.

Is it the end of the 90 minutes and are you wondering if the rest can be done next time?

Yes

Thank God. Please ask the question and let's get out of here!

No

Ok, you have a legitimate question. Do you actually care about the answer?

Not really, I just want to show off.

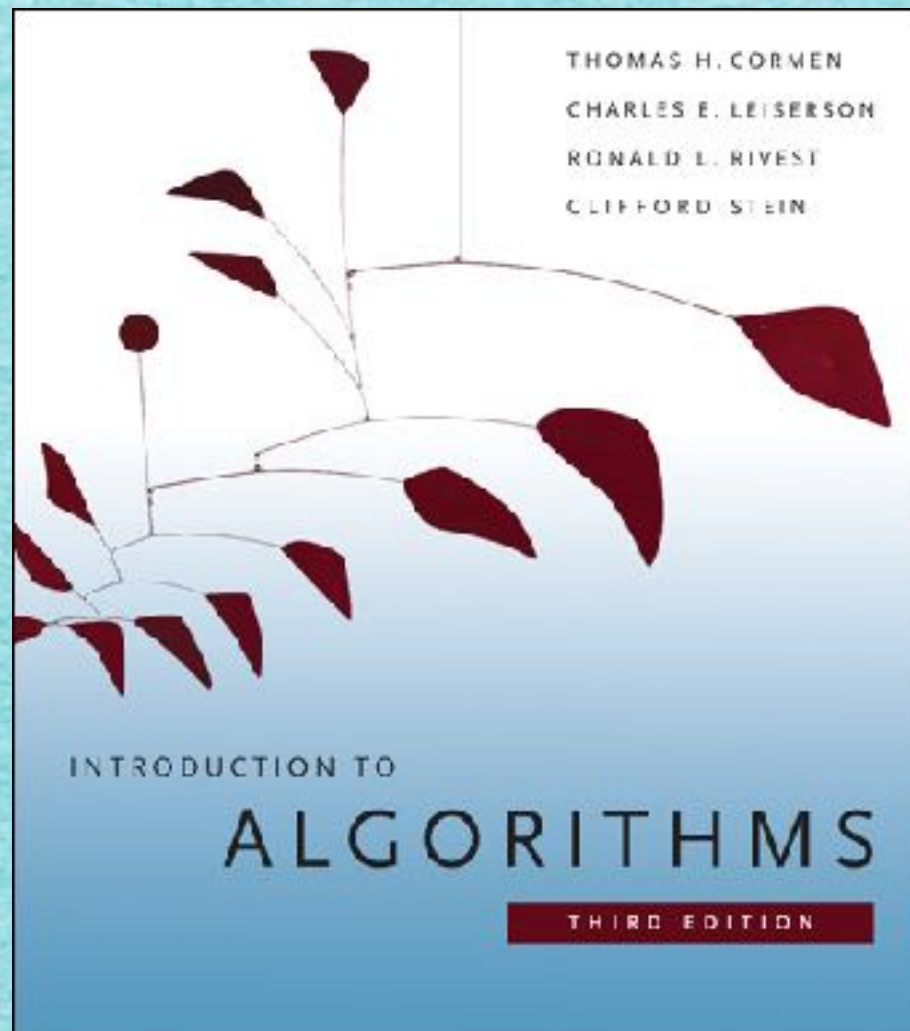
Yes!

FINE, ASK YOUR QUESTION.

JORGE GIMM © 2013

WWW.PHDCOMICS.COM

Literatur



Google



1.1 Was ist ein Algorithmus?



“Ein **Algorithmus** ist eine aus endlich vielen Schritten bestehende eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen.”

Beispiele:

- Kochrezept
- Bedienungsanleitung
- Notenblatt
- Programmablaufplan

1.1 Was ist ein



“Ein
einde

Beispiele:

-
-
-
-

Schritten bestehende
es Problems oder einer

HOT DOG

The diagram illustrates the assembly of a hot dog in seven steps. At the top, the ingredients are listed: 2x buns, 1x hot dog, 4x rings, 1x mustard, and 1x relish. The steps are: 1. Open the bun. 2. Place the hot dog in the bun. 3. Add the rings. 4. Add relish. 5. Add mustard. 6. Add mustard. 7. Close the bun.

IKEA Restaurant & Café 9.00am - 9.30pm

1.1 Was ist ein Algorithmus?



“Ein **Algorithmus** ist eine aus endlich vielen Schritten bestehende eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen.”

Beispiele:

- Kochrezept
- Bedienungsanleitung
- Notenblatt
- Programmablaufplan

Kein Computer ohne Algorithmen!
Keine Informatik ohne Algorithmen!
Kein Informatikstudium ohne Algorithmen!

Wie rechnet man richtig?



Muhammad al-Chwarizmi (783-850)



Al-gabr
(„Algebra“)

Wie rechnet man richtig?



$XLII + XCVIII = ?$

Muhammad al-Chwarizmi (783-850)

Wie rechnet man richtig?



Muhammad al-Chwarizmi (783-850)

$XLII + XCVIII = ?$

$42 + 98 = 140!$

Wie rechnet man richtig?



$XLII + XCVIII = CXL$

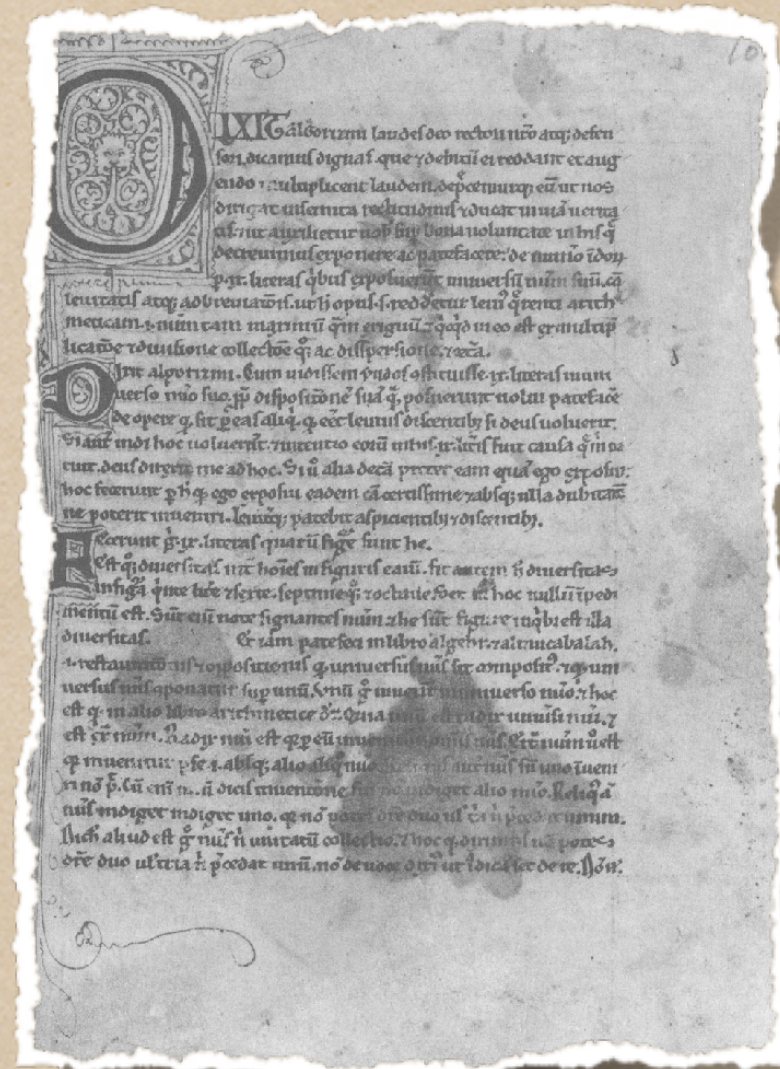
$42 + 98 = 140!$

Muhammad al-Chwarizmi (783-850)

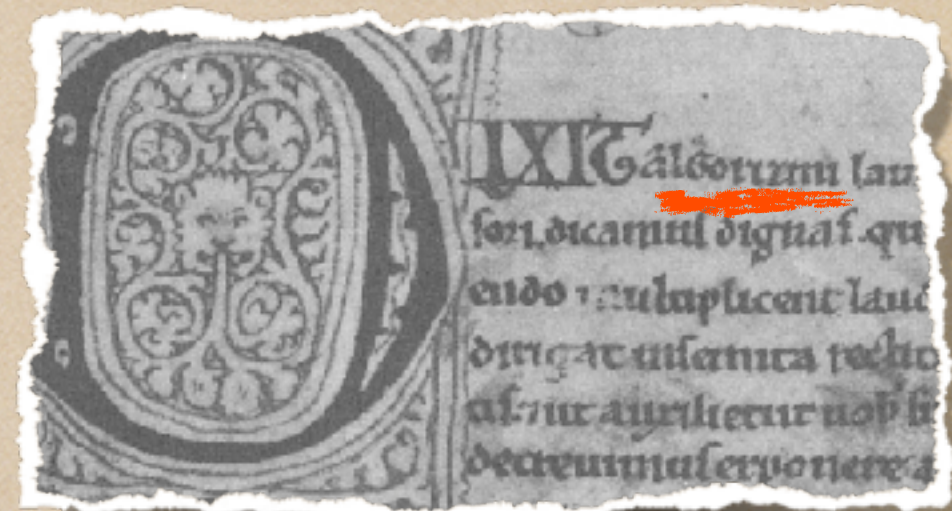
Wie rechnet man richtig?



Muhammad al-Chwarizmi (783-850)



Wie rechnet man richtig?



Muhammad al-Chwarizmi (783-850)

Algorithmus: Eine systematische Rechenvorschrift zum Lösen eines Problems in endlich vielen Schritten

Datenstrukturen

SMBC SATURDAY MORNING
BREAKFAST CEREAL
BY ZACH WEINERSMITH

Datenstrukturen

SMBC

SATURDAY MORNING
BREAKFAST CEREAL
BY ZACH WEINERSMITH

HALF OUR DATA?
ON WHAT BASIS?



DAMN. DAMN,
THAT'S PRETTY
GOOD.



Datenstrukturen

SOME CENTURIES PRIOR...

I CALL
IT ZERO.



1.1 Was ist ein Algorithmus?

“Ein **Algorithmus** ist eine aus endlich vielen Schritten bestehende eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen.”

Die Wikipedia-Definition gibt einem zunächst einmal eine einigermaßen anschauliche Vorstellung, gerade wenn man eine sehr genaue Vorstellung davon hat, was man für ein Problem lösen möchte und wie eine Lösung aussieht. Allerdings sind die Begriffe etwas vage...

Also:

- Was ist ein “Problem”?
- Was ist “eine Klasse von Problemen”?
- Was ist “die Lösung eines Problems”?

1.1 Was ist ein Algorithmus?

Konkretes Beispiel:

- (1) Bestimme den größten gemeinsamen Teiler von 144 und 729!
- (2) Bestimmt den größten gemeinsamen Teiler von x und y !

Lösung:

- (1) 9
- (2) Euklidischer Algorithmus

1.1 Was ist ein Algorithmus?

**Also: Nicht verwirren lassen von den verschiedenen Ebenen!
Dabei hilft saubere Schreibweise und Sprachgebrauch.**

1.1 Was ist ein Algorithmus?

Problem 1.1 (*Größter gemeinsamer Teiler*)

Gegeben: Zwei positive ganze Zahlen, x und y

Gesucht: Die größte ganze Zahl z mit der Eigenschaft, dass z sowohl x als auch y teilt

Eine derartig allgemein beschriebene Aufgabenstellung bezeichnen wir als Problem.

Eine Aufgabenstellung mit konkreten Zahlen bezeichnen wir als Instanz eines Problems.

1.1 Was ist ein Algorithmus?

Die Lösung eines Problems besteht nun in einem systematischen Verfahren, das einem für jede beliebige Instanz eine Lösung dieser Instanz liefert.

Gegeben: Eine Problembeschreibung

Gegeben: Eingabedaten

Gesucht: Ausgabedaten



Gesucht: Ein Algorithmus, der das Problem löst

1.2 Wie formalisiert man einen Algorithmus?

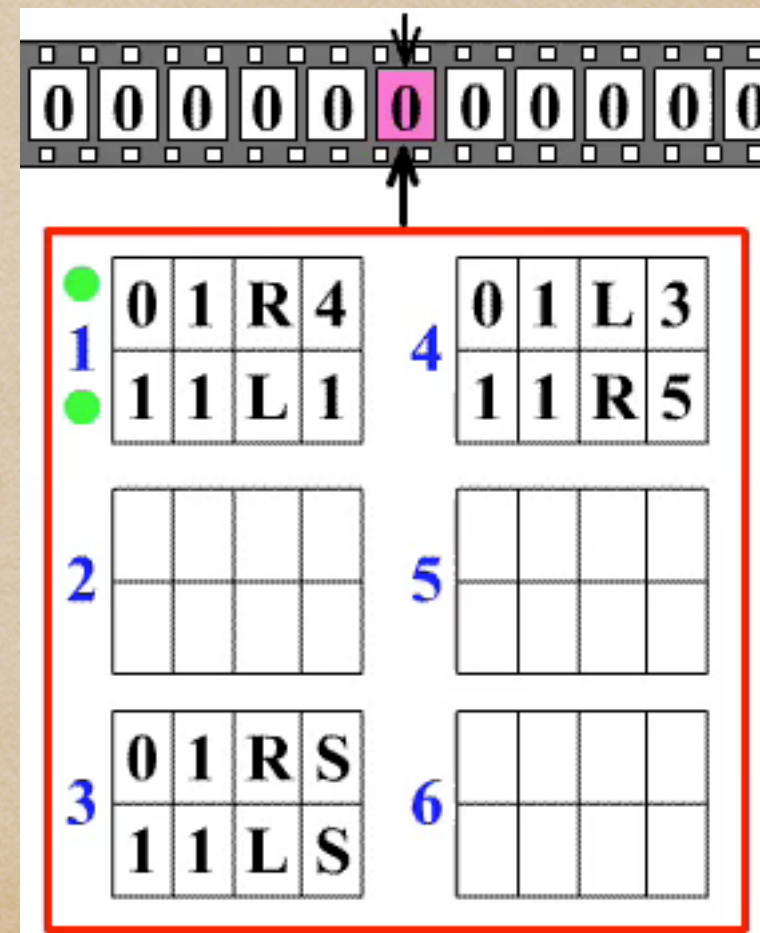


Alan Turing (1912-1954)

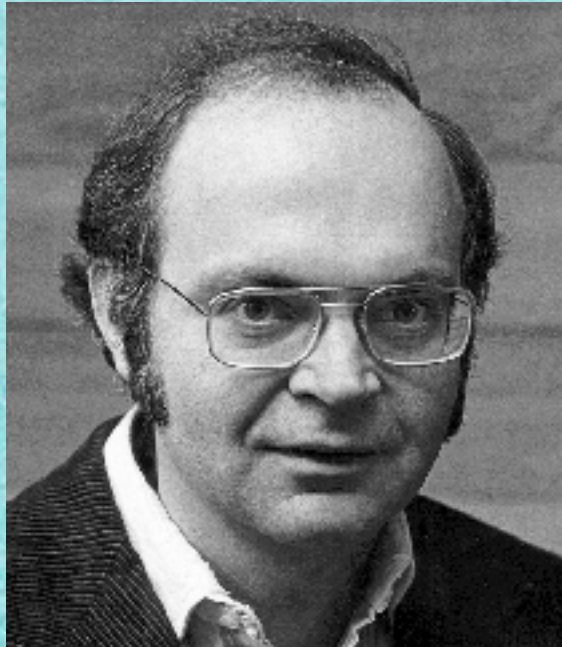
1.2 Wie formalisiert man einen Algorithmus?



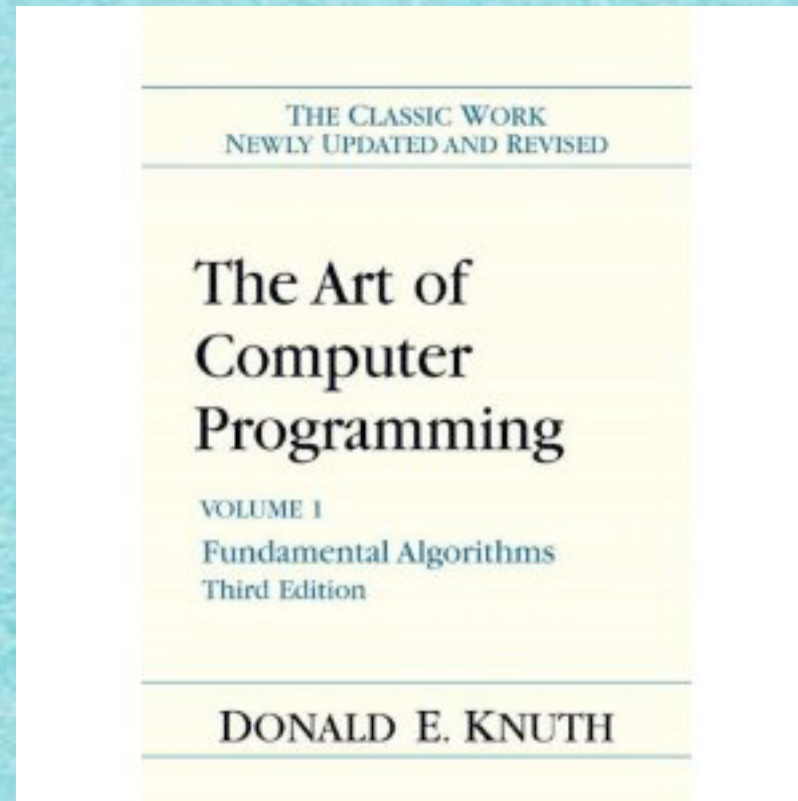
Idee: Abstrakte Formalisierung beliebiger Berechnungen!



1.3 Eigenschaften von Algorithmen

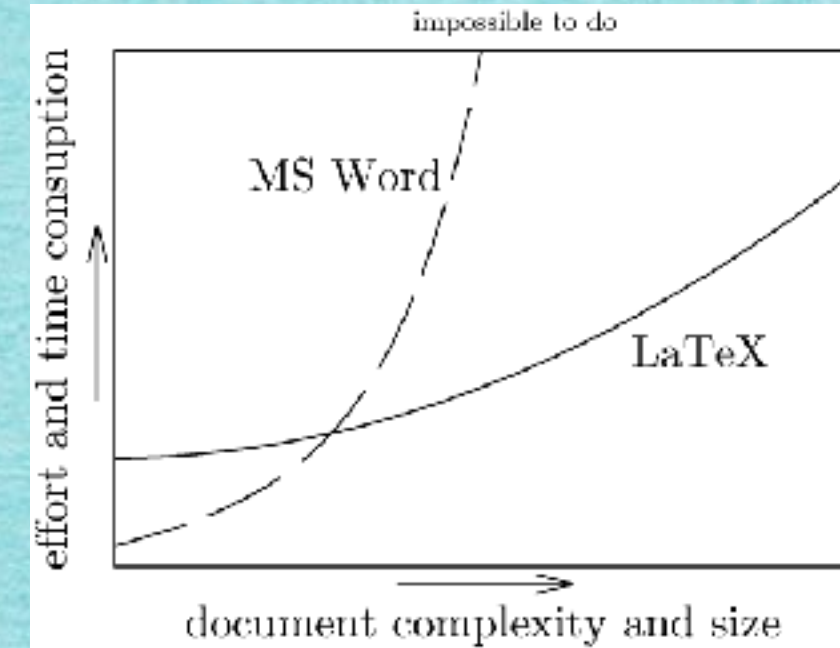
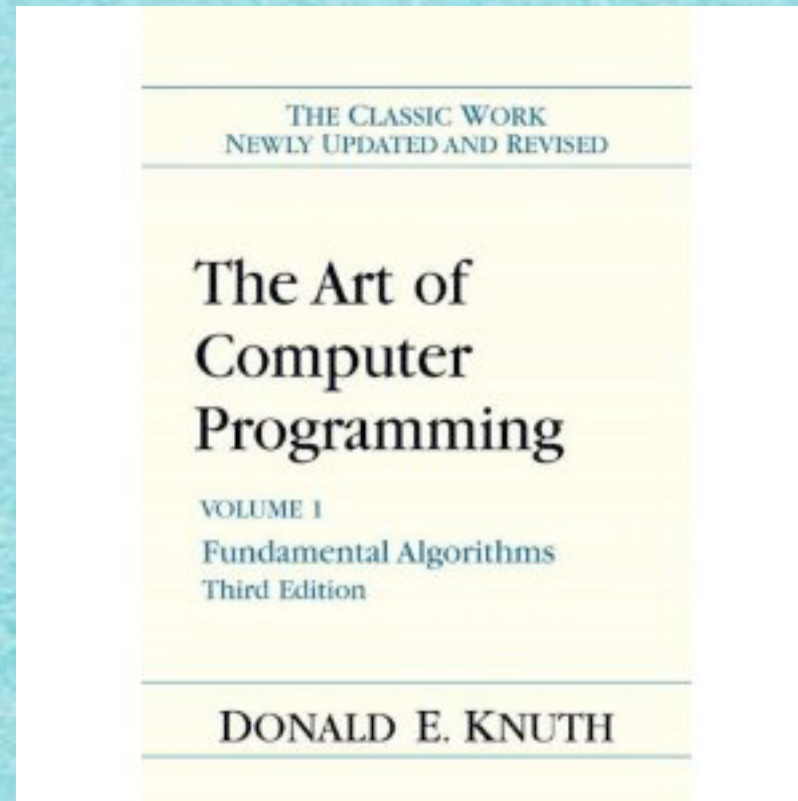


1.3 Eigenschaften von Algorithmen



Donald Knuth (1938-)

1.3 Eigenschaften von Algorithmen

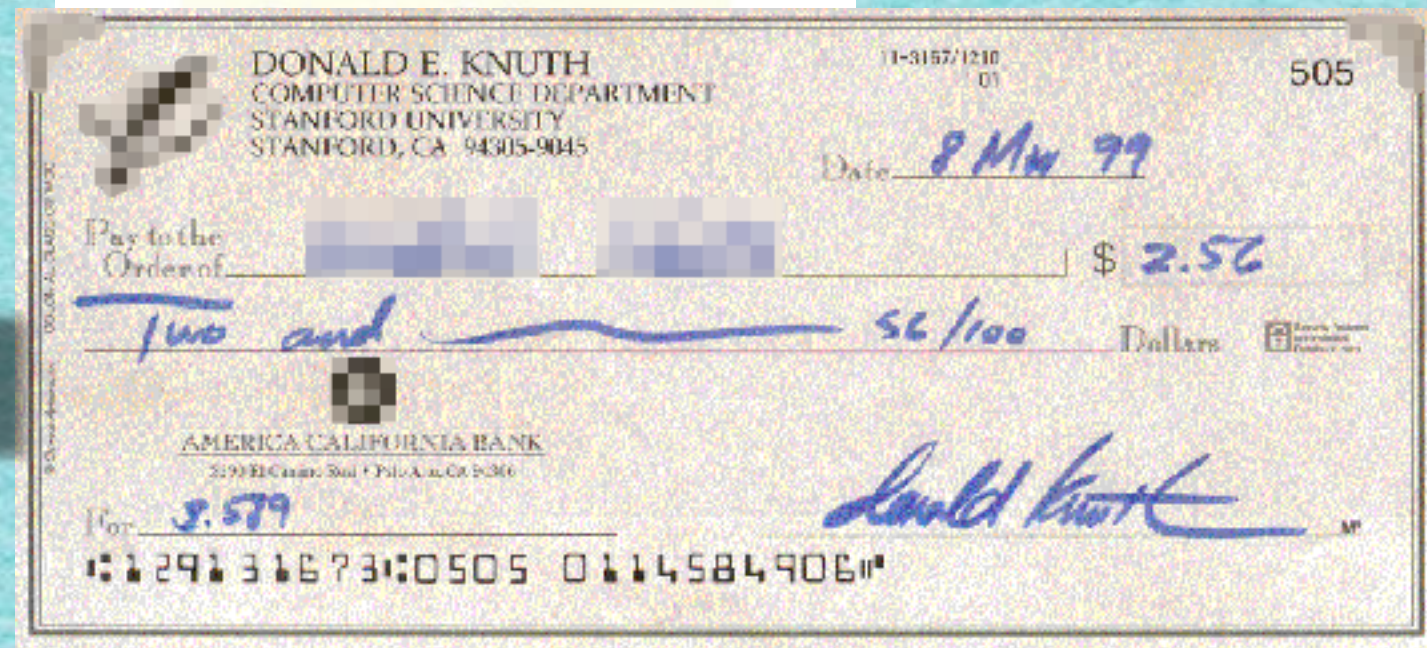
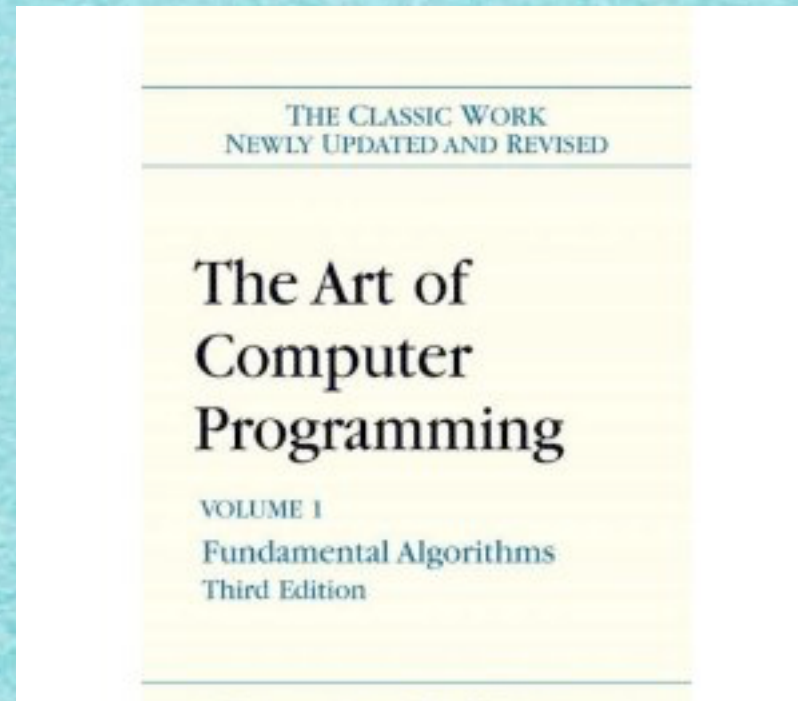


Donald Knuth (1938-)

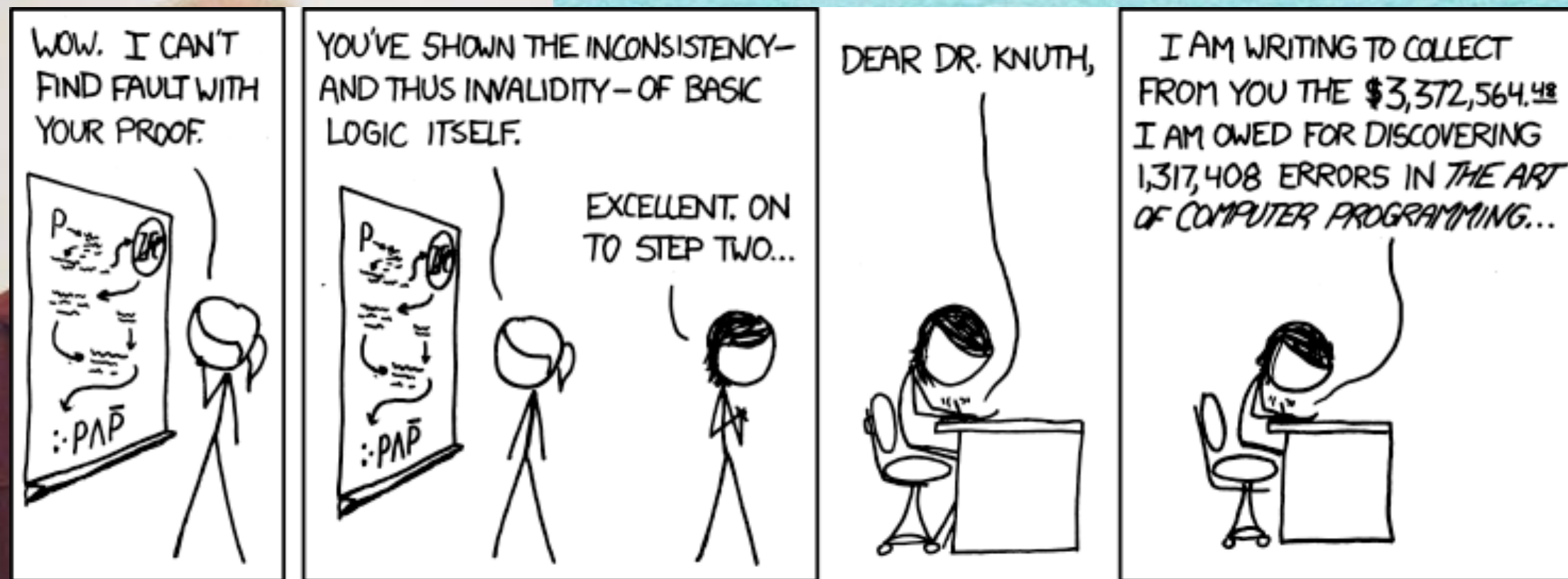
1.3 Eigenschaften von Algorithmen



Donald Knuth (1938-)



1.3 Eigenschaften von Algorithmen

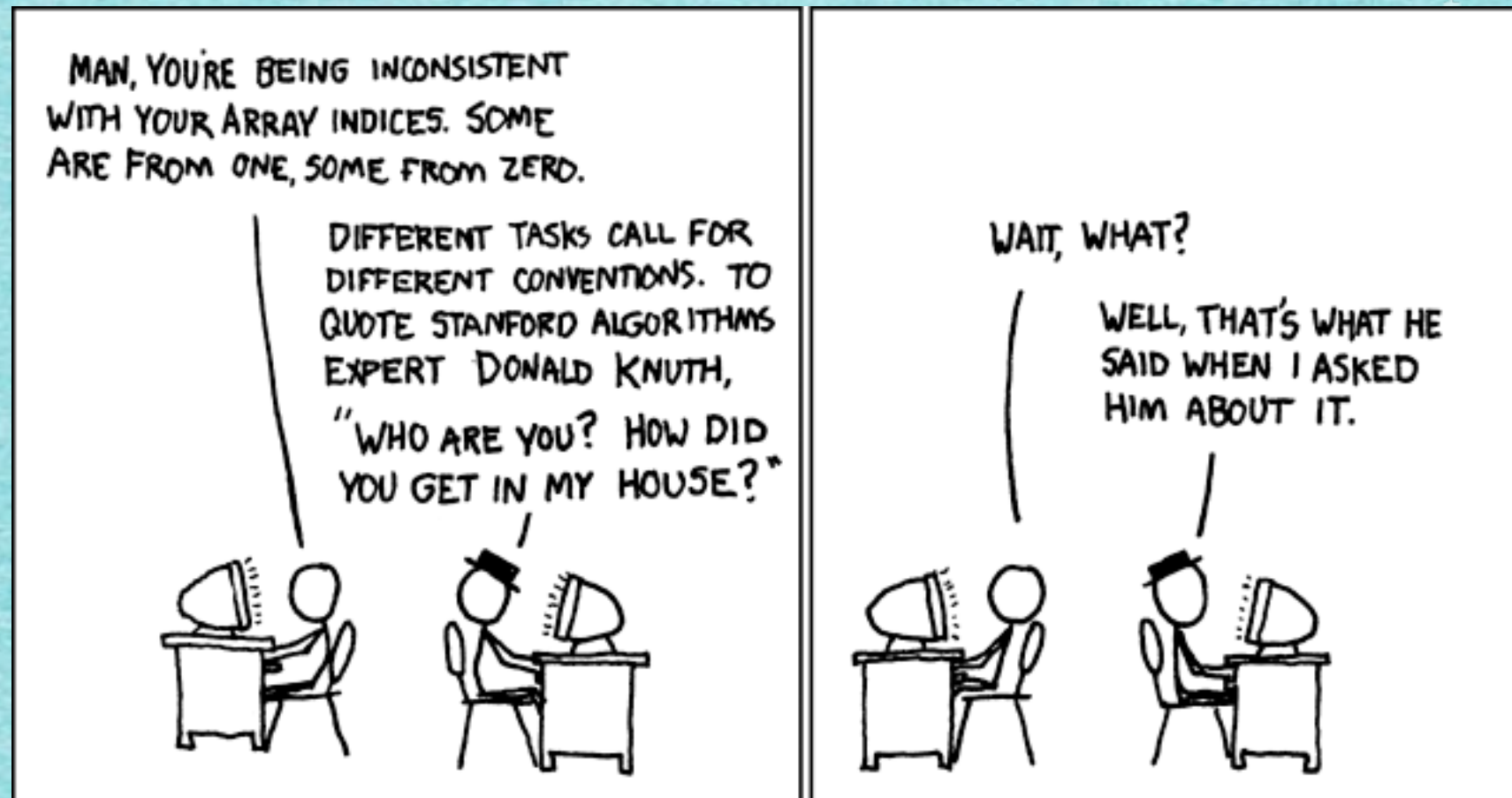


Donald Knuth (1938-)

1.3 Eigenschaften von Algorithmen



Donald Knuth (1938-)



1.3 Eigenschaften von Algorithmen

Eine Berechnungsvorschrift zur Lösung eines Problems heißt genau dann Algorithmus, wenn eine zu dieser Berechnungsvorschrift äquivalente Turingmaschine existiert, die für jede Eingabe, die eine Lösung besitzt, stoppt.

1.3 Eigenschaften von Algorithmen

1. Das Verfahren muss in einem endlichen Text eindeutig beschreibbar sein (Finitheit).
2. Jeder Schritt des Verfahrens muss tatsächlich ausführbar sein (Ausführbarkeit).
3. Das Verfahren darf zu jedem Zeitpunkt nur endlich viel Speicherplatz benötigen (Dynamische Finitheit, siehe [Platzkomplexität](#)).
4. Das Verfahren darf nur endlich viele Schritte benötigen ([Terminierung](#), siehe auch [Zeitkomplexität](#)).

1.3 Eigenschaften von Algorithmen

Determiniertheit: Gleicher Input liefert gleiches Ergebnis.

Determinismus: Alle Schritte sind genau festgelegt.

1.3 Eigenschaften von Algorithmen

Determiniertheit: Gleicher Input liefert gleiches Ergebnis.

Determinismus: Alle Schritte sind genau festgelegt.

Randomisierter Algorithmus: Man darf “würfeln”.

1.4 Datenstrukturen

Eine Datenstruktur erlaubt es, die für eine Aufgabe notwendigen Informationen

- geeignet zu repräsentieren
- den Zugriff und die Verwaltung während der Bearbeitung in effizienter Weise zu ermöglichen.

Mehr Einzelheiten werden uns ab dem dritten Kapitel begegnen!

$$XLII + XCVIII = CXL$$

$$42 + 98 = 140!$$

1.5 Ausblick

Wir werden uns in dieser Vorlesung mit verschiedenen Aspekten von Algorithmen beschäftigen. Dazu gehören oft auch Analyse und Verständnis der zugrundeliegenden mathematischen Struktur. Gerade letzteres macht oft den eigentlichen Witz aus!

1.5 Ausblick

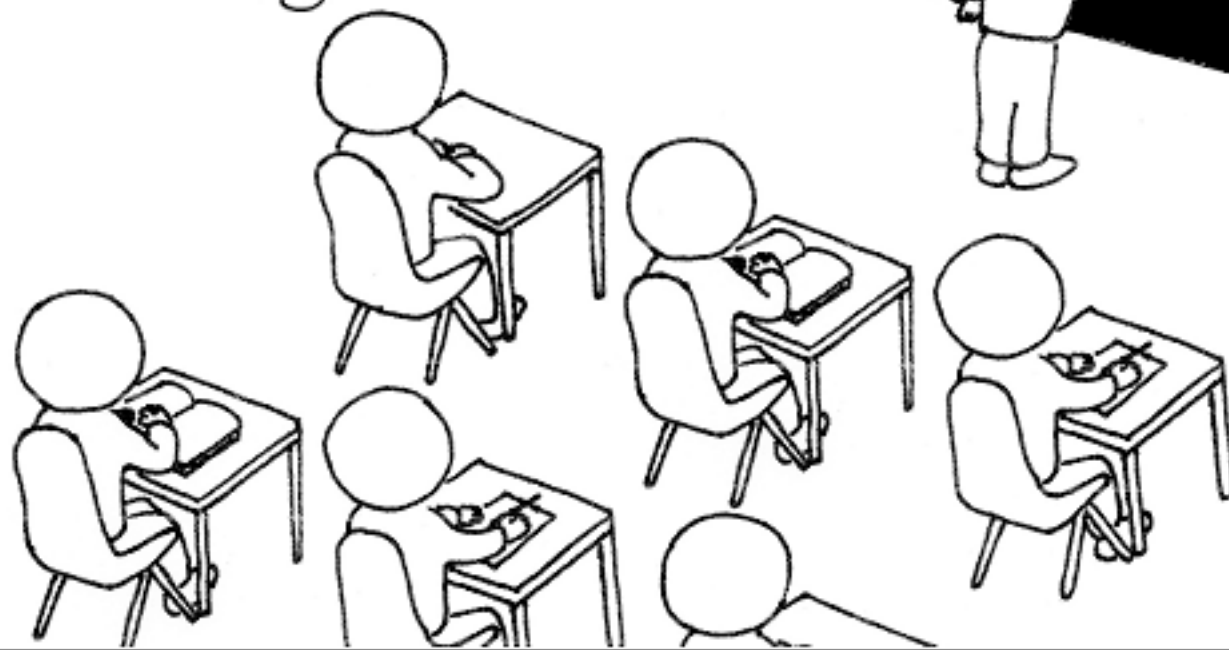
Dieses Verständnis ist wichtig um

- ein Gefühl für die Besonderheiten eines Problems zu bekommen.
- ein Gefühl für das Funktionieren einer Lösungsmethode zu bekommen.
- eine Lösungsmethode zu entwickeln.
- zu zeigen, dass eine Lösungsmethode korrekt ist.
- zu zeigen, dass es keine Lösung gibt.
- zu zeigen, dass es keine Lösungsmethode gibt.
- **SPASS DABEI ZU HABEN!**

THUS, FOR ANY NONDETERMINISTIC TURING MACHINE M THAT RUNS IN SOME POLYNOMIAL TIME $p(n)$, WE CAN DEVISE AN ALGORITHM THAT TAKES AN INPUT w OF LENGTH n AND PRODUCES $E_{M,w}$. THE RUNNING TIME IS $O(p^2(n))$ ON A MULTITAPE DETERMINISTIC TURING MACHINE AND...

WTF, MAN. I JUST WANTED TO LEARN HOW TO PROGRAM VIDEO GAMES.

SIPSER CH7
 $y_{i,j-1,0} \wedge y_{i,j,0} \wedge y_{i,j,1,0} \wedge y_{i,j,1,1}$
 $y_{i,j-1,0} \wedge y_{i,j,0} \wedge y_{i,j,1,0} \wedge y_{i,j,1,1}$
 $N_i = (A_{i0} \vee B_{i0}) \wedge (A_{i1} \vee B_{i1}) \wedge \dots \wedge$
 $N = N_0 \wedge N_1$



Vielen Dank!

s.fekete@tu-bs.de