Ideen der Informatik

Suchen und Sortieren

[Ordnung muss sein…]

Kurt Mehlhorn

Adrian Neumann

viele Folien von Kostas Panagiotou

Suchen

- Welche Telefonnummer hat Kurt Mehlhorn?
- Wie schreibt man das Wort “Equivalenz”?
- Welche Webseiten enthalten die Wörter “Uni Saarland”? Welche ist die “wichtigste”?

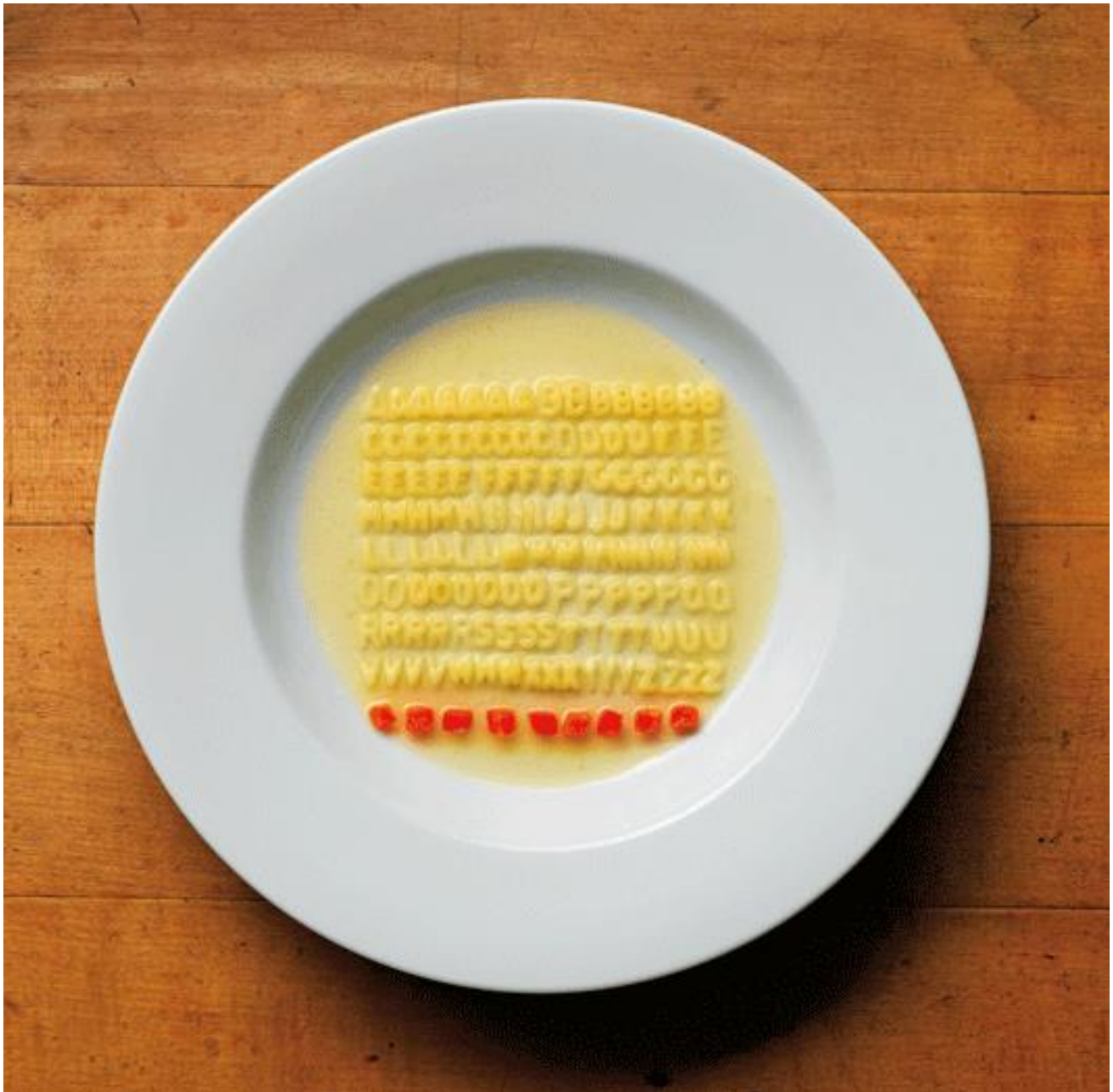
Web hat mehrere Billionen Seiten: Suche nach der Nadel im Heuhaufen.

Bedeutung von Suchen

- Menschen verbringen viel Zeit mit Suchen und Ordnen (Sortieren), Computer auch
- Suchen und Sortieren sind Hauptanwendungen von Computern
- Es gibt hocheffiziente Suchverfahren: Suche im Web in weniger als 1 Sekunde
- Sortieren hilft beim Suchen: Ordnung ist das halbe Leben

Gibt es ein X in der Buchstabensuppe?

Aha!



Ein grünes Badetuch?

Aha!

Bilder von
Ursus Wehrli, 2011

Konzepte der heutigen Vorlesung

- Suchen = Information ablegen und verarbeiten, so dass man sie schnell wiederfindet (oder sagen kann, das sie nicht vorhanden ist)
- Ordnung erleichtert das Suchen
- Sortieren = nach einem Kriterium ordnen
- Datenstrukturen = Suchen in Mengen, die sich zeitlich ändern
- Laufzeit (Komplexität) von Algorithmen

Aufgaben

- Nach welcher Regel sind die Namen in einem Telefonbuch geordnet? Man nennt diese Ordnung die alphabetische Ordnung

Suchen

- Daten können alles Mögliche sein:
  - Zeichenketten, Zahlen, Bilder, …

- Hier: (Name + Telefonnummer)
- Haben einen Karteikasten: auf jeder Karteikarte steht ein Name und eine Nummer
- Wie viele Zettel muss man anschauen, bis man die Nummer zu einem Namen hat?

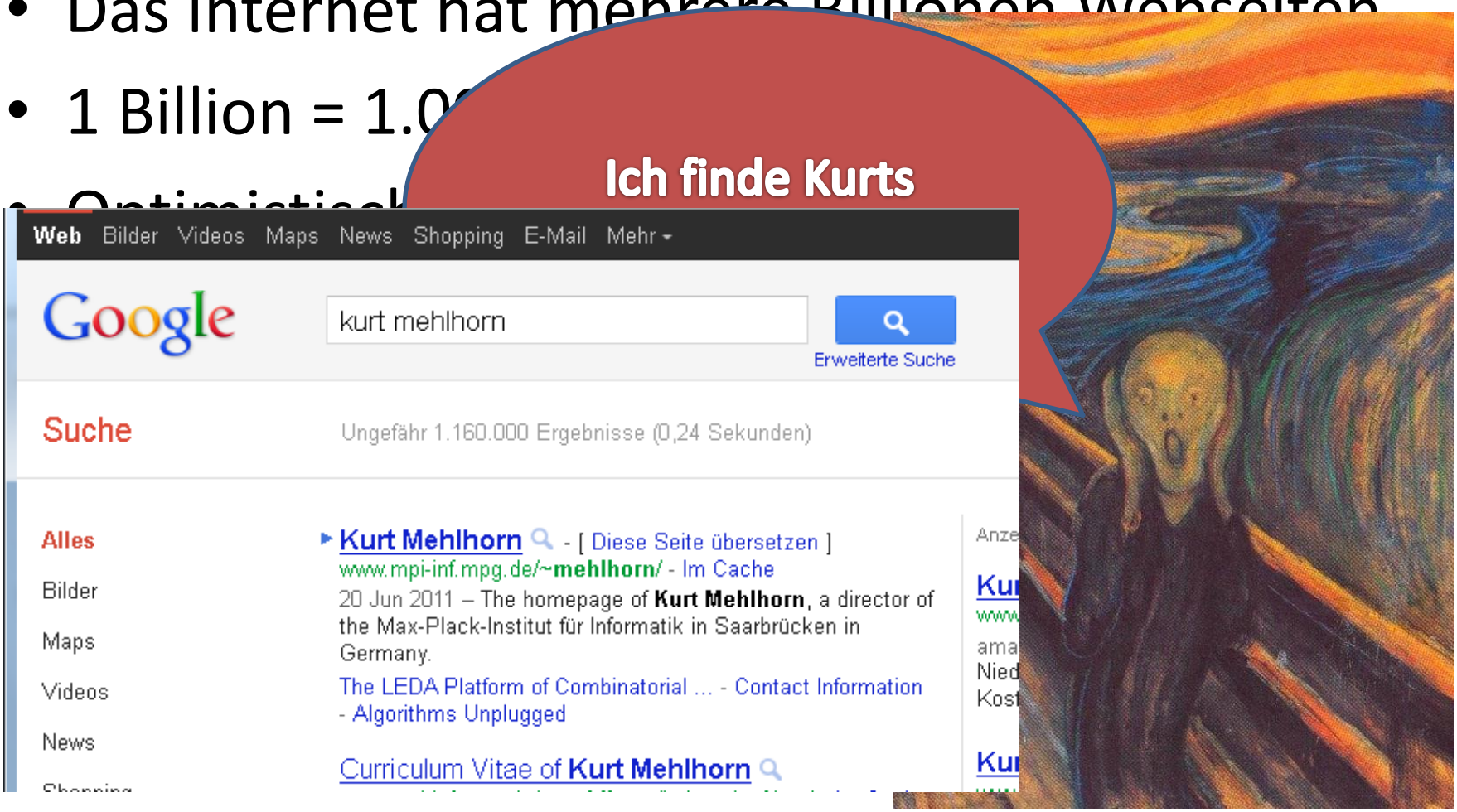
Zettel sind ungeordnet

- Wir müssen **alle Karten** anschauen, um sicher zu sein, dass ein gesuchter Name nicht da ist.
- Falls ein Name da ist, im Mittel die Hälfte der Karten

Anzahl der Vergleiche im schlechtesten Fall = Anzahl der Karten

Ein Beispiel

- Das Internet hat mehrere Billionen Webseiten
- 1 Billion = 1.0
- Optimistisch



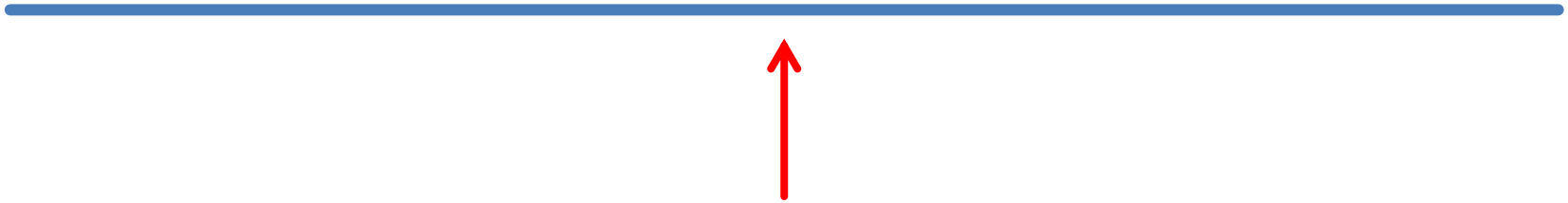
Wie machen wir das besser?

- Wir sortiern unsere Karteikarten nach Name
  - Also wie in einem Telefonbuch
- Wir suchen nach X
- Wir ziehen eine Karte, darauf steht Y
- X kommt vor/nach Y in der alphabetischen Ordnung der Namen oder X= Y
- Was wissen wir nun? Welche Karte nehmen wir für den ersten Vergleich? Wie geht es weiter?

Binärsuche

- Gegeben:
  - Liste mit ***N*** Elementen
  - Sortiert: der Nachfolger i
    Vorgaenger.
- Frage: enthaelt die Liste ein Element ***x***?
- Algorithmus:

Konzept: Divide and Conquer



Der Algorithmus

- Karteikasten: L[1],L[2],…,L[N] N Karten

Suche(L[1], …, L[N], x)

falls N = 0 dann fertig, x nicht vorhanden

Sei *m* das mittlere Element in L, also *m* = L[N/2]

falls x = *m* dann fertig, x ist gefunden

falls x < *m* dann Suche(L[1], …, L[N/2-1], x)

falls x > *m* dann Suche(L[N/2+1], …, L[N], x)

Komplexität (Anzahl der Vergleiche)

- N = 1: Vergleiche = 1
- N = 3: Vergleiche = 2
- N = 7: Vergleiche = 3
- N = 15: Vergleiche = 4
- N = 31: Vergleiche = 5

2^{40} = 1.099.511.627.776

- N = 15 = 2x2x2x2 – 1 = 2^4 - 1 Vergleiche = 4
- N = 31 = 2x2x2x2x2 – 1 = 2^5 - 1 Vergleiche = 5
- N = …. = 2^{40} - 1 Vergleiche = 40

Das ist irre!!!!!

- Anzahl Vergleiche = Zweierlogarithmus(N + 1)

Lineare Suche vs. Binärsuche

- Binärsuche funktioniert wenn man die gegebenen Daten ordnen kann:
  - (Name, Telefonnummer)
  - Webseiten?
- ***Lineare Suche: Aufwand = Anzahl der Elemente***
- ***Binärsuche: Aufwand = Logarithmus der Anzahl der Elemente***
- ***Binärsuche ist rasend schnell***

Aufgabe

- Wir suchen nach einem Element x in einer Menge von n = 2^k - 1 Elementen. Ein Vergleich zwischen zwei Elementen dauert 1 Sekunde. Wie lange dauert die Suche bei Verwendung von linearer Suche bzw. Binärsuche für k = 2, k = 10, k = 20, k = 30?

Wie sortiert man?

Mischen zweier sortierter Folgen

- Zwei aufsteigend sortierte Folgen von je n Elementen kann man mit höchstens 2n – 1 Vergleichen zu einer sortierten Folge mischen
- Strategie: vergleiche die beiden ersten Elemente und bewege das kleinere zur Resultatfolge

Mischen Pseudocode

Seien A[0] bis A[n-1] und B[0] bis B[n-1] sortierte Folgen. Stelle C[0] bis C[2*n – 1] für das Ergebnis bereit

Setze i und j auf Null

Solange (i $<$ n oder j $<$ n)

Falls (i $<$ n und j $<$ n und A[i] $<$ B[j]) oder j $==$ n

dann bewege A[i] nach C[i + j] und erhöhe i

sonst bewege B[j] nach C[i + j] und erhöhe j

Sortieren durch Mischen

- Wir haben n = 2^k Elemente und damit n sortierte Folgen der Länge 1
- Paare die sortierten Folgen und mische je zwei zu einer Folge der doppelten Länge.
- Solange noch mehr als eine Folge, wiederhole

Sortieren durch Mischen, Pseudocode

In A[0] bis A[n-1] steht die Eingabe; n = 2^k; wir benutzen auch noch B[0] bis B[n-1]

Setze L auf 1; // A besteht aus n/L sortierten Folgen der Länge L; beginnen bei 0,L,2*L,3*L, …

Solange n/L > 1

    Für i = 0, 2*L, … , n/(2*L) – 1

        Mische A[i]…A[i+L–1] und A[i+L]…A[i+L+L–1] und schreibe das Ergebnis nach B[i] bis B[i+2*L-1];

    Kopiere B nach A;

    Verdopple L;

Sortieren durch Mischen, Analyse

1. Wir haben n = 2^k Elemente und damit n sortierte Folgen der Länge 1
2. Paare die sortierten Folgen und mische je zwei zu einer Folge der doppelten Länge.
3. Solange noch mehr als eine Folge, wiederhole

4. Jede Ausführung von 2) kostet nicht mehr als n Vergleiche.
5. Wir machen 2) k-mal.
6. Also nicht mehr als n k = n log n Vergleiche.

Sortieren durch Mischen

sortiert n Elemente

mit nicht mehr als n log n Vergleichen

Was bedeutet das für praktisch?

Tatsächliche Laufzeit auf KMs Rechner

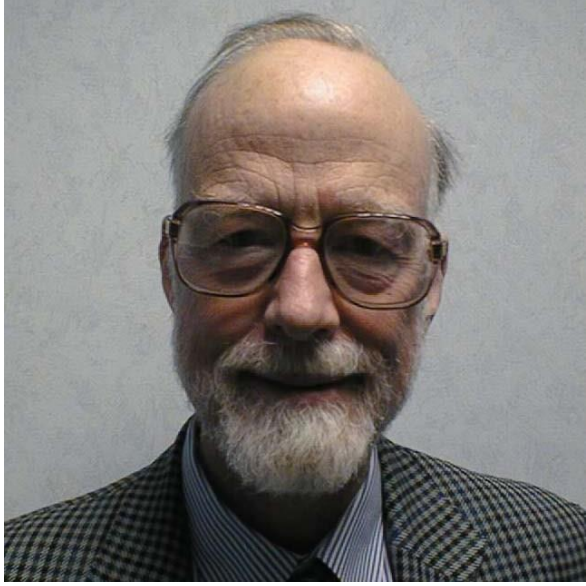
- n = 2^{22} 1.09 seconds
- n = 2^{25} 9.94 seconds
- n = 2^{29} 183 seconds
- n = 2^{30} 1240 Sekunden

- Beachte 2^{29} log 2^{29} / 2^{25} log 2^{25} = 16 $\frac{29}{25}$ = 18.56
- 183 / 9.94 = 18.41
- Analyse sagt Laufzeitwachstum gut vorher
- Aber letzte Zeile: Rechner ist in anderem Regime, benutzt Platte

Quicksort

- S = Menge, die zu sortieren ist
- Wähle ein Element s in S
- Teile S in
  - $S_<$ = Elemente kleiner s
  - $S_>$ = Elemente größer s
- Gib aus

$$\text{Sort}(S_<) \;\; s \;\; \text{Sort}(S_>)$$



Tony Hoare
Turing Award 1980

Rekursion endet, wenn S< und S> leer sind

Quicksort, Beispielausführung

Beim Teilen kann man Glück oder Pech haben

Laufzeit wie $n \log n$	n^2
$n = 10^6$	0.1 sec	500 sec

Kann man das Glück erzwingen?

- Bis 1980: immer raffiniertere deterministische Strategien
- Seit 1980: wähle das Teilungselement zufällig

  Randomisierter Algorithmus

- Urne mit n/2 roten und n/2 schwarzen Kugeln. Wie findet man eine rote Kugel ohne hinsehen?

Zusammenfassung Sortieren

- Sortieren geht recht schnell, eine Million Elemente in 0.1 Sekunden auf Notebook

- Weltrekorde
  - Eine Billion Zahlen in drei Stunden
  - 50 Milliarden Zahlen für einen Penny

Tony Hoare (1934 --

„Ich stelle fest, dass es zwei Wege gibt, ein Software-Design zu erstellen, entweder so einfach, dass es offensichtlich keine Schwächen hat, oder so kompliziert, dass es keine offensichtlichen Schwächen hat. Die erste Methode ist weitaus schwieriger.“

Tony Hoare, Dankesrede für den Turingpreis 1980

„I think Quicksort is the only really interesting algorithm that I've ever developed. “

Zusammenfassung

- Binärsuche ist rasend schnell: 40 Vergleiche für Suche in einer Billion Elemente
- Sortieren ist billig: eine Million Elemente in 0.1 sec auf diesem Notebook
- Suchbäume erlauben Binärsuche auf dynamischen Daten

Aufgaben

- Wieviele Vergleiche macht Mergesort höchstens, um $n = 2^{25}$ Elemente zu sortieren.
- Sei S eine Menge von n = 2k + 1 Elementen. Der Median von S ist ein Element x in S, so dass k Elemente größer sind als x und k Elemente kleiner. Ändern sie Quicksort so ab, dass es den Median bestimmt.
- Sei S eine Menge von n Elementen. Sie möchten rausfinden, ob die Elemente paarweise verschieden sind.
  - Methode A: jedes Element mit jedem anderen vergleichen
  - Methode B: Sortieren und dann …