15.

Heaps

Buch Mark Weiss „Data Structures & Problem Solving Using Java“ siehe:
- 833-852 (Heap, Heapsort)

Lernziele Kapitel 15 Heaps

- Heap-Datenstruktur und ihre Implementierung verstehen
- Die Operationen insert und get_min angeben können
- Heapsort verstehen
- Zeitkomplexität Heapsort / Heap-Operationen begründen können

Thema / Inhalt

Heaps sind eine etwas verrückte (aber effiziente und sehr nützliche) Datenstruktur, da muss man erst mal draufkommen! In gewisser Weise stellen sie „halbwegs" sortierte Strukturen dar: Wenn man etwas entfernt oder einfügt, dann muss man nicht viel Aufwand spendieren, damit die Struktur danach auch wieder halbwegs sortiert ist. Und wenn man das kleinste oder das grösste Element haben möchte, oder wenn man gar den ganzen Inhalt in sortierter Weise geliefert bekommen möchte, dann geht das auch recht effizient, weil ja schon alles „halbwegs" sortiert vorliegt.

Heaps eignen sich tatsächlich in idealer Weise zum Sortieren (**Heapsort**): Alles in irgendeiner Reihenfolge in den Heap einfügen, dann wiederholt das kleinste Element daraus entfernen. Beides geht bei einem Heap recht effizient. Nur: Wenn man dem Algorithmus bei der Arbeit zuschaut und das Verfahren nicht schon kennt, dann versteht man rein gar nichts – es scheinen

Thema / Inhalt (2)

einfach irgendwelche Elemente wild hin- und hergetauscht zu werden! Wir gehen die Sache aber systematisch an, wodurch sich die Funktionsweise offenbart.

Im Sinne einer **abstrakten Datenstruktur** realisiert ein Heap eine sogenannte „**priority queue**" – ein „Behälter", der Elemente mit einem geordneten Schlüsselwert (z.B. vom Typ int oder float) speichern kann und zwei Operationen anbietet: „insert", womit ein Element in den Behälter eingefügt wird, und „get_min", wodurch das Element mit dem kleinsten Schlüsselwert der im Behälter derzeit gespeicherten entfernt und ausgeliefert wird.

Heaps sind dabei als Binärbäume organisiert, bei denen alle inneren Niveaus vollständig gefüllt sind, an der Wurzel das kleinste Element steht und alle Pfade von der Wurzel zu einem Blatt aufsteigend sortiert sind. Wie wir sehen werden, kann dann sowohl „insert" als auch „get_min" in logarithmischer Zeit (bezogen auf die Gesamtzahl der Elemente) erfolgen. Vor allem dann, wenn man Heaps niveauweise in einem Array speichert, können die beiden Operationen sehr effizient ausgeführt werden – sie verwirren aber einen unbedarften Beobachter, der die Interpretation als Binärbaum nicht erkennt.

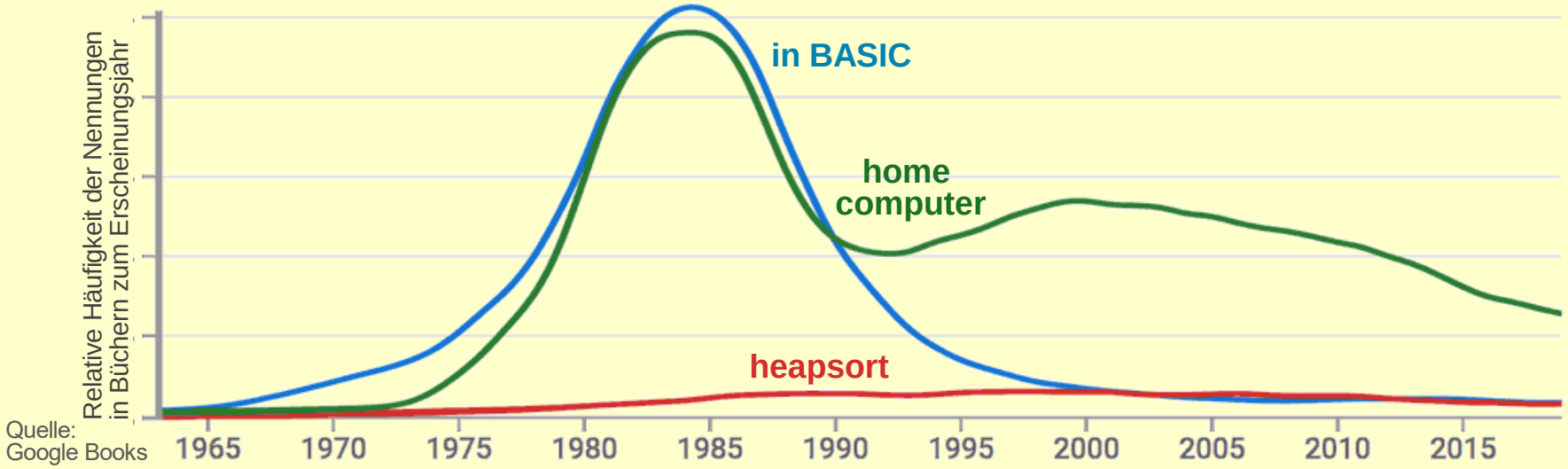
Heaps lassen sich verschiedentlich anwenden. Zum Beispiel qualifizieren sie sich als Datenstruktur für die Planungsliste bei der ereignisgesteuerten Simulation: Dynamisch entstehende Ereignisnotizen, die in Zukunft auszuführen sind, werden entsprechend ihres Eintrittszeitpunktes eingefügt; zyklisch wird vom Simulator die kleinste Ereignisnotiz entnommen und zur Ausführung gebracht. Häufig ist auch der Einsatz in Vorrangwarteschlangen, wie sie bei Servern oder Betriebssystemen zur Festlegung der Ausführungsreihenfolge von Aufgaben benötigt werden.

Java bietet mit **java.util.PriorityQueue** Heaps als direkt nutzbare Datenstruktur an.

Thema / Inhalt (3)

Heapsort wurde 1964 entdeckt und veröffentlicht. Im selben Jahr wurde die Welt mit der Programmiersprache **BASIC** beglückt: Eine gegenüber Algol, Fortran und Cobol deutlich einfachere, für Anfänger geeignete Programmiersprache, die vor allem interaktiv auf den bald danach aufkommenden Heim- und Hobbycomputern genutzt werden konnte und auf einem Laufzeitinterpreter statt einem Compiler beruhte, was die eigentliche Ausführung zwar verlangsamte, aber das System einfacher und direkter anwendbar machte. In unserem historischen Strang zeigen wir eine Implementierung von Heapsort in BASIC und vermitteln einen kurzen Eindruck davon, wie seinerzeit mit den Heimcomputern, die noch nicht das Prädikat „PC" trugen, programmiert wurde.

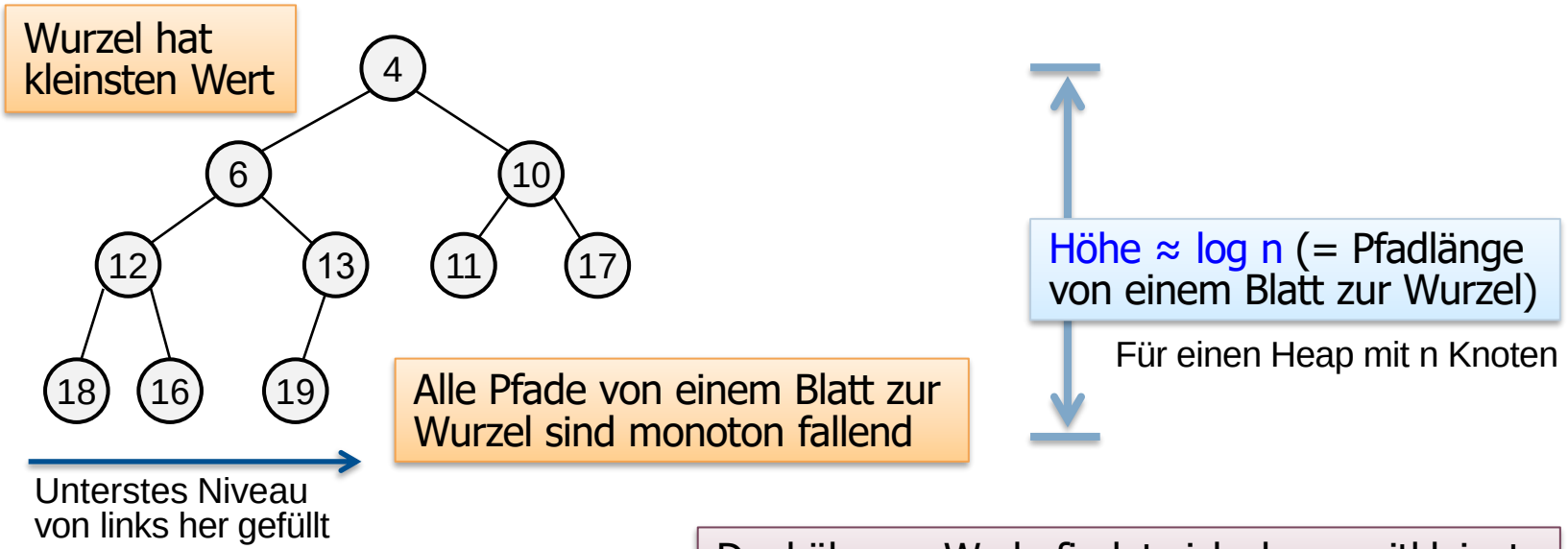
Ein **visueller Wettlauf** verschiedener Sortierverfahren und eine **Vertonung von Heapsort** beschliessen das Kapitel.



Die Heap-Datenstruktur

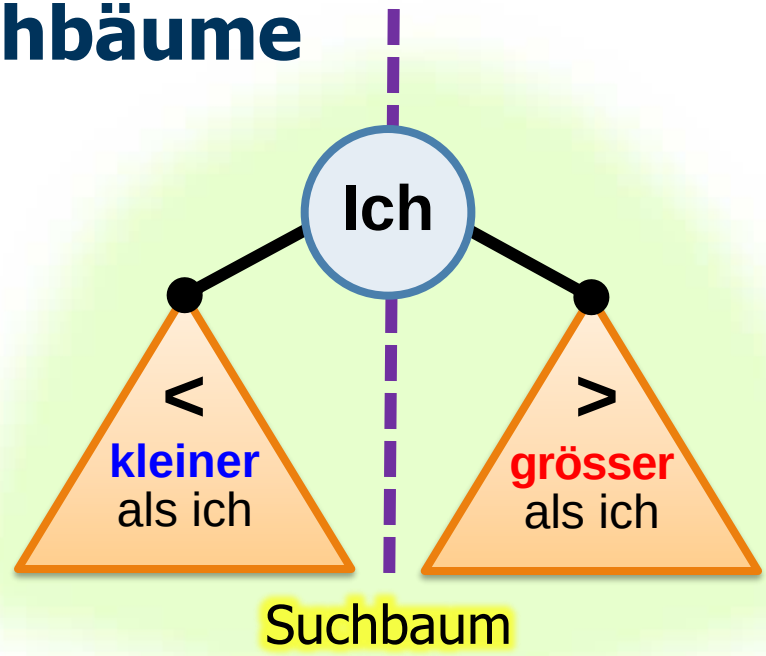
- Heap = Binärbaum, für den gilt:
  - Er ist (fast) vollständig
  - Knoten besitzen einen Wert
    (wir nehmen hier an, dass dieser eindeutig ist, die Koten also unterschiedliche Werte haben)
  - Für alle Knoten k ≠ Wurzel: Wert(Vorgänger(k)) < Wert(k)
    (alternative Def. mit ≤, >, ≥ auch möglich)

Min-Heap

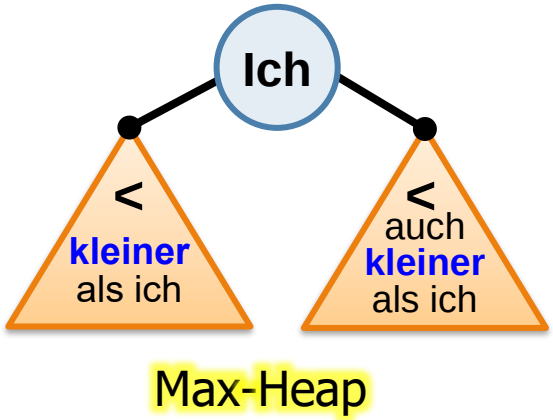


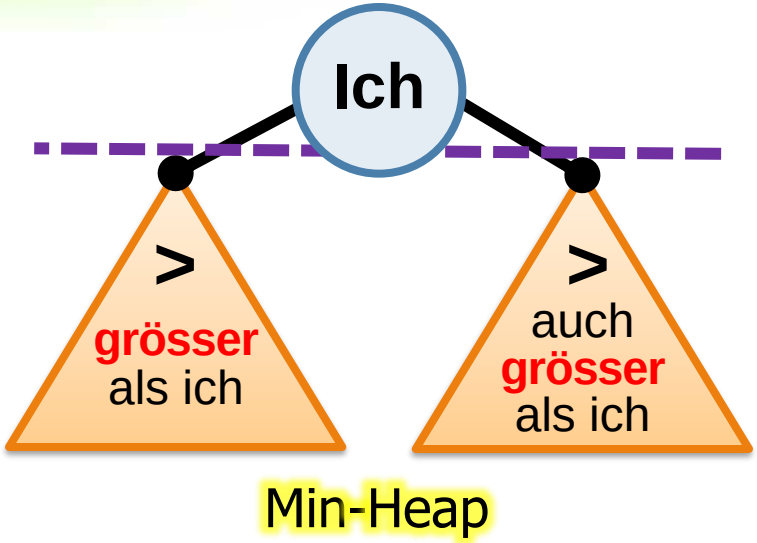
Denkübung: Wo befindet sich das zweitkleinste Element in einem Heap? Und wo das grösste?

Heaps ⇔ Suchbäume



Beachte auch:
Ein x-beliebiges Element im linken Unterbaum ist bei einem *Suchbaum* kleiner als ein y-beliebiges Element des rechten Unterbaums. Eine solche Eigenschaft bzgl. der Unterbäume gilt aber nicht bei den „schwächer" geordneten *Heaps*!



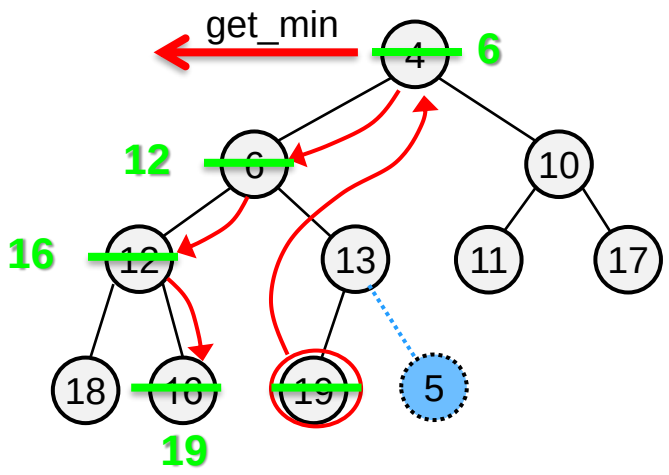


Heap-Operationen: get_min und insert

Man überlege sich genau, dass das der Fall ist!

Für get_min *und* insert gilt:
- Lässt Heap-Eigenschaft invariant
- Benötigt O(log n) Schritte

Im Unterschied zur Implementierung von priority queues als sortierte / unsortierte verkettete Liste!

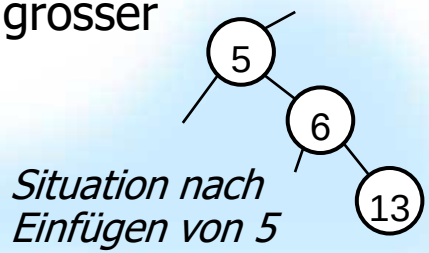


get_min:

- Wurzel (hier: 4) entfernen
- Letzten Knoten des untersten Niveaus (hier: 19) an die Wurzelposition setzen
- Neue Wurzel so weit wie möglich „herunterbubbeln" lassen: Mit kleinerem der beiden Nachfolger vertauschen; dies rekursiv (oder iterativ) auf entsprechenden Unterbaum anwenden

insert:

- Als neues nächstes Blatt (hier: 5) auf unterstem Niveau einfügen
- „Hochbubbeln": So weit wie möglich nach oben wandern lassen – iterativ mit Vorgänger vertauschen, wenn dieser grösser

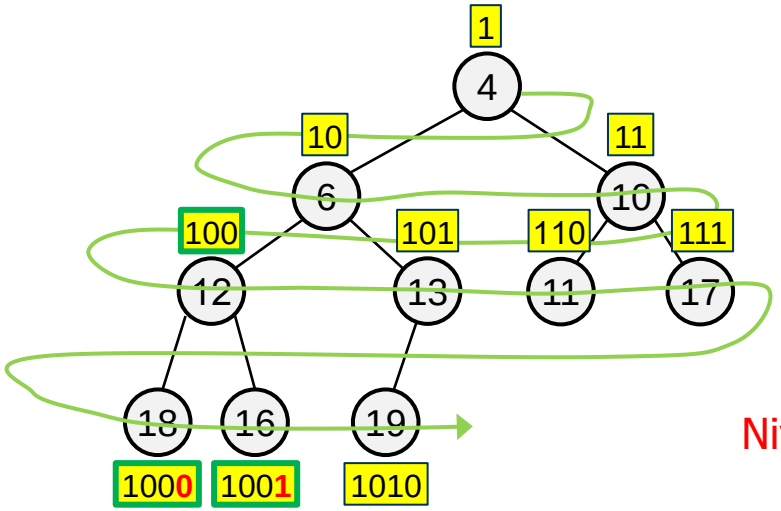


Situation nach Einfügen von 5

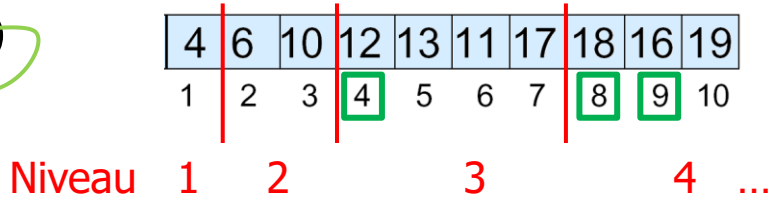
Heaps niveauweise als Array

Vgl. dazu frühere Slide „Binärbäume in Arrays“

- Wurzel steht bei Array-Index 1
- Direkte Nachfolger eines Knotens mit Index i haben die Indizes 2i und 2i+1



Beispiel für einen Heap, der solcherart „niveauweise“ gespeichert ist



- Damit ist Aufsteigen / Absteigen entlang eines Astes bei get_min / insert besonders einfach!
  - Halbieren / Verdoppeln von Indizes

PARENT (i)
→ return $\lfloor i/2 \rfloor$
LEFT_CHILD (i)
→ return $2i$
RIGHT_CHILD (i)
→ return $2i+1$

Heap: Implementierung von „insert“
(bei niveauweiser Speicherung des Heaps in einem int-Array „a“)

Als neues Blatt auf unterstem Niveau einfügen.

So weit wie möglich nach *oben wandern* lassen: Iterativ mit Vorgänger vertauschen, wenn dieser grösser.

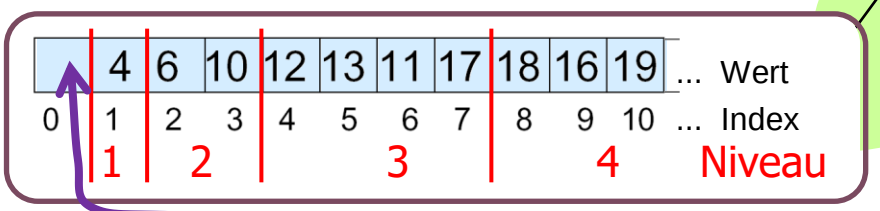
```
void insert (int x) {
  int k = ++N;
  a[0] = -1;
  while (a[k/2] >= x) {
    a[k] = a[k/2];
    k = k/2;
  }
  a[k] = x;
}
```

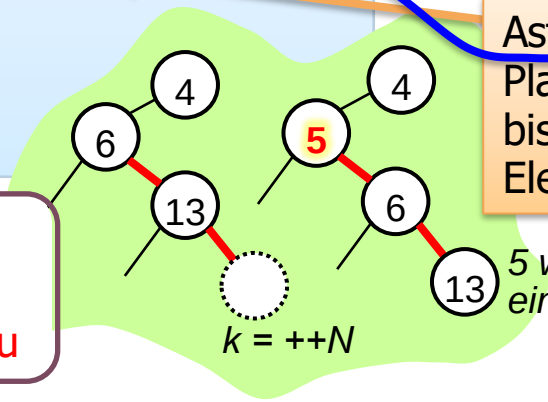
N bezeichnet die Anzahl der gespeicherten Elemente

Ein „Trick“, damit die Schleife *immer* verlassen wird

k/2 wird evtl. abgerundet

Ast hochwandern, alles einen Platz nach unten schieben, bis die Stelle für das neue Element gefunden ist





5 wurde eingefügt

Denkübung: (1) Kann man „>“ statt „>=“ verwenden? (2) Was geschieht, wenn man ein schon gespeichertes Element einfügt?

- Trick mit „superkleinem“ Wert -1 bei a[0] klappt so nur, wenn keine negativen Werte eingefügt werden!
- Beachte: Bei Ersetzen von k/2 durch k-1 erhält man insertion sort
  ⇒ Interpretation: Beim Heap wird insertion sort entlang eines einzigen Astes angewandt; dieser hat hier aber nur eine Länge von ≈ log n

Heap: „get_min“

```
int get_min() {
  int k = 1; int j; int x = a[1];
  a[1] = a[N--];
  int w = a[1];
  while (k <= N/2) {
    j = k + k;
    if (j<N && a[j+1] < a[j])
      j++;
    if (w <= a[j])
      break;
    a[k] = a[j];
    k = j;
  }
  a[k] = w;
  return x;
}
```

Letzter Knoten wird neue Wurzel

Shortcut! (Wieso nicht „&“ ?)

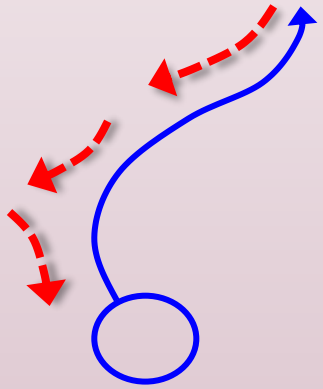
Platz k gefunden: kleiner als beide Nachfolger (falls vorhanden)

Wurzel w hin-absin-ken lassen

Beispiel für j und k

k	1	3	6	12
j	3	6	12	25

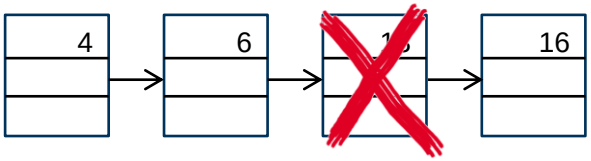
Wurzel entfernen, danach letzten Knoten des untersten Niveaus an die Wurzelposition setzen.

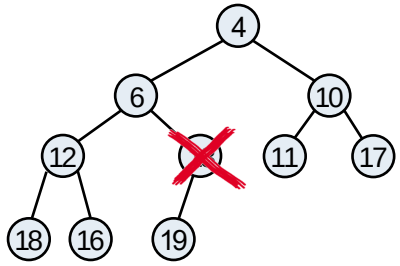


Neue Wurzel so weit wie möglich nach *unten sinken* lassen: Iterativ mit kleinerem der beiden Nachfolger vertauschen…

Denkübung: Was geschieht bei Anwendung auf einen leeren Heap?

Heap: „cancel“





- Manchmal ist eine cancel-Operation praktisch
  - Ein gewisses früher eingefügtes Element x („vorzeitig“) herausnehmen
- Beispiele (bei Priority-Queues):
  - Reisebüro-Szenario: Bei rechtzeitiger Bedienung die (nur für den Eventualfall vorgemerkte) Ereignisnotiz für „Geduld-Ende“ canceln
  - Druckauftrag annullieren bzgl. einer Drucker-Warteschlange
  - Entfernen eines „timer events“, das einem nur im seltenen Notfall (z.B.: Kommunikationspartner antwortet lange nicht) erlösen soll
- Lösungsidee („Anti-Heap“):
  - Element x nicht wirklich entfernen, sondern für ungültig erklären
  - Ein Anti-Heap verwaltet die Ungültigkeitsnotizen; das zeitlich nächste ungültige Element steht dort immer an der Wurzel
  - cancel(x): x in den Anti-Heap einfügen
  - get_min modifizieren: wenn das zurückgelieferte Element x auch das nächste Element im Anti-Heap ist, dann ist es ungültig → x aus beiden Heaps (an der Wurzel) entfernen und get_min rekursiv aufrufen

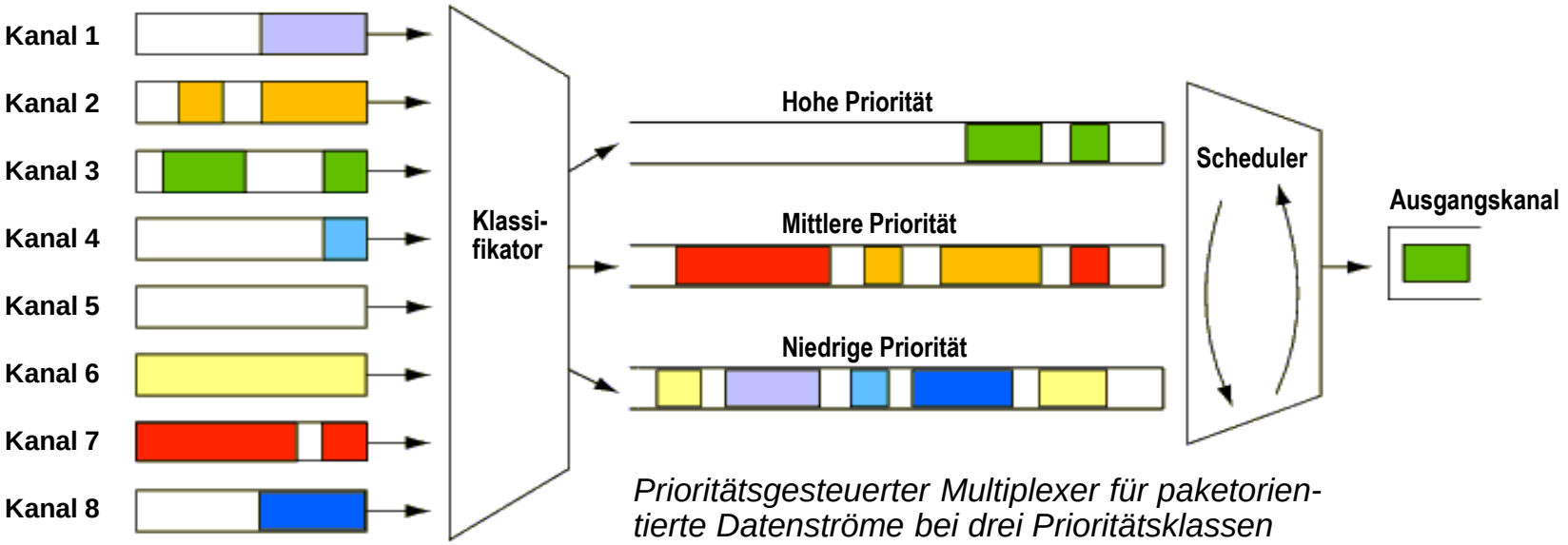
Nutzung von Priority Queues

- Simulator-Ereignislisten
- Betriebssysteme, Kommunikationssysteme
  - Scheduling: Prozess / Thread mit *höchster* Priorität als nächsten
  - Routing: Datenpakete *hoher Priorität* (z.B. für Interaktion) bevorzugen
- Spielbaumanalyse, kombinatorische Optimierung
  - *Besten* Zug zuerst analysieren
  - *Aussichtsreichstes* Teilproblem zuerst bearbeiten
- Codierungstheorie, Dateikomprimierung
  - Kurze Codewörter für *häufigste* Zeichen („Huffman-Code")

Wir rufen ein früheres Problem in Erinnerung: Eignen sich Heaps (besser / schlechter – und in welcher Hinsicht?) zur Implementierung der dynamischen Terminplanungsliste (Bsp.: „Reservierung von Landezeitpunkten?")

Priority Q's für wenige feste Prioritätsklassen

- Dann evtl. wie am Flughafen
  - Pro Prioritätsklasse eine eigene FIFO-Warteschlange
  - Auswahl („dequeue") prioritär aus höherrangierter Teilwarteschlange



Prioritätsgesteuerter Multiplexer für paketorientierte Datenströme bei drei Prioritätsklassen

Heapsort

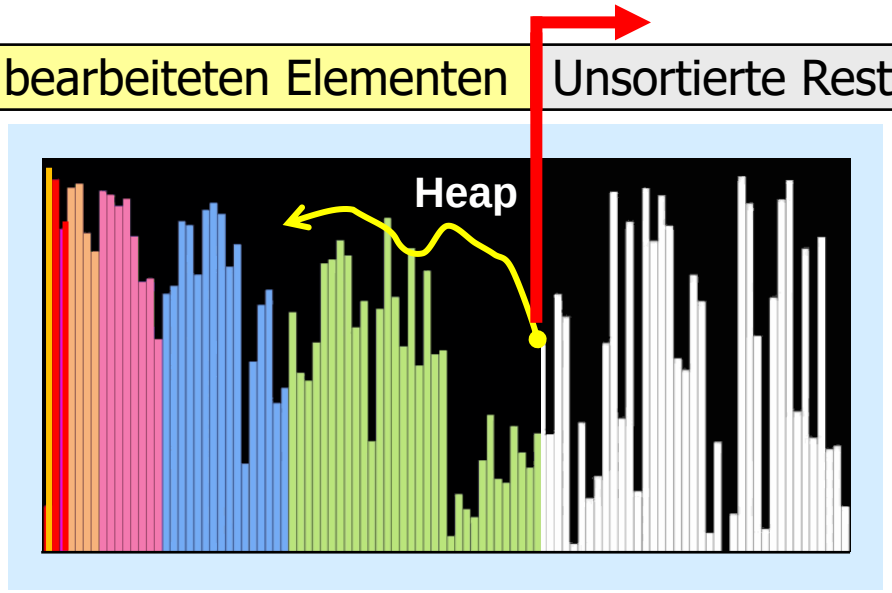
Beide Phasen fügen jeweils n Elemente ein bzw. aus, was jeweils log n Einzelschritte benötigt (die Astlänge ist nur „kurzzeitig" wesentlich kleiner als log n); jede Phase hat daher O(n log n) Aufwand, zusammen daher ebenfalls O(n log n)

- Sortieren mit einem Heap:
  1) n Elemente (aus unsortierter Ausgangsfolge) nacheinander einfügen
  2) danach n mal *get_min* auf den Heap anwenden

  ⇒ Sortierverfahren mit Zeitkomplexität O(n log n) im worst case (und best case)

- Man kann den Heap im Array selbst („in situ", ohne zusätzlichen Platzbedarf) aufbauen, indem dort der Heap von links heranwächst, während nacheinander Elemente des unsortierten (rechten) Teils entfernt werden:

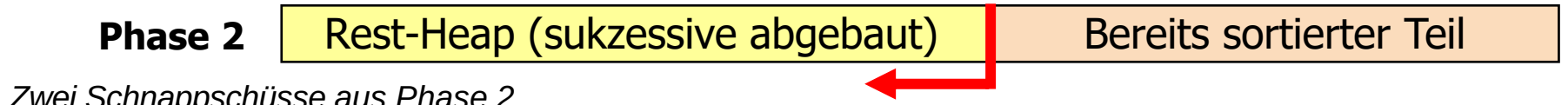
Phase 1 | Heap aus bereits bearbeiteten Elementen | Unsortierte Resteingabe



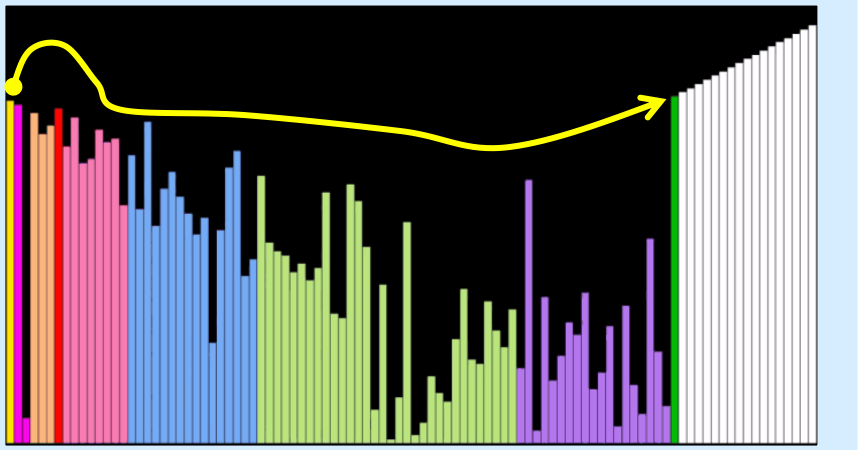
Schnappschuss aus Phase 1 (Hier mit einem Max-Heap)

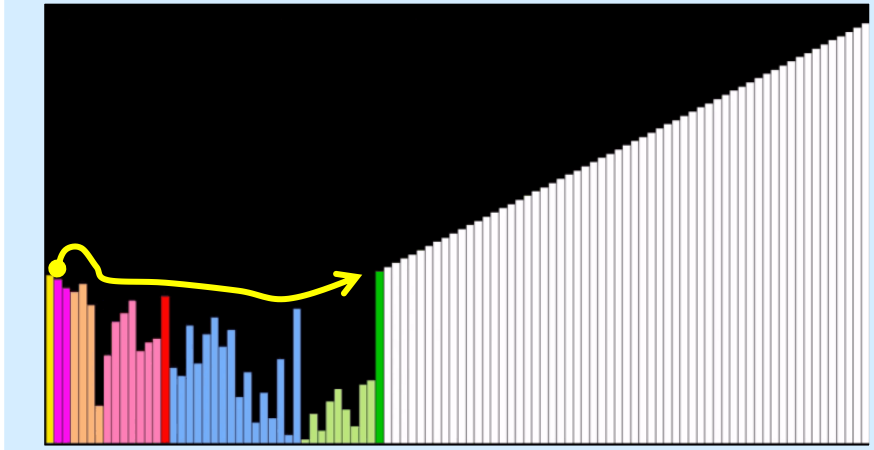
Heapsort (2)

Danach wird der Heap schrittweise abgebaut und das jeweils entfernte Element an die im rechten Teil heranwachsende sortierte Folge links angefügt:



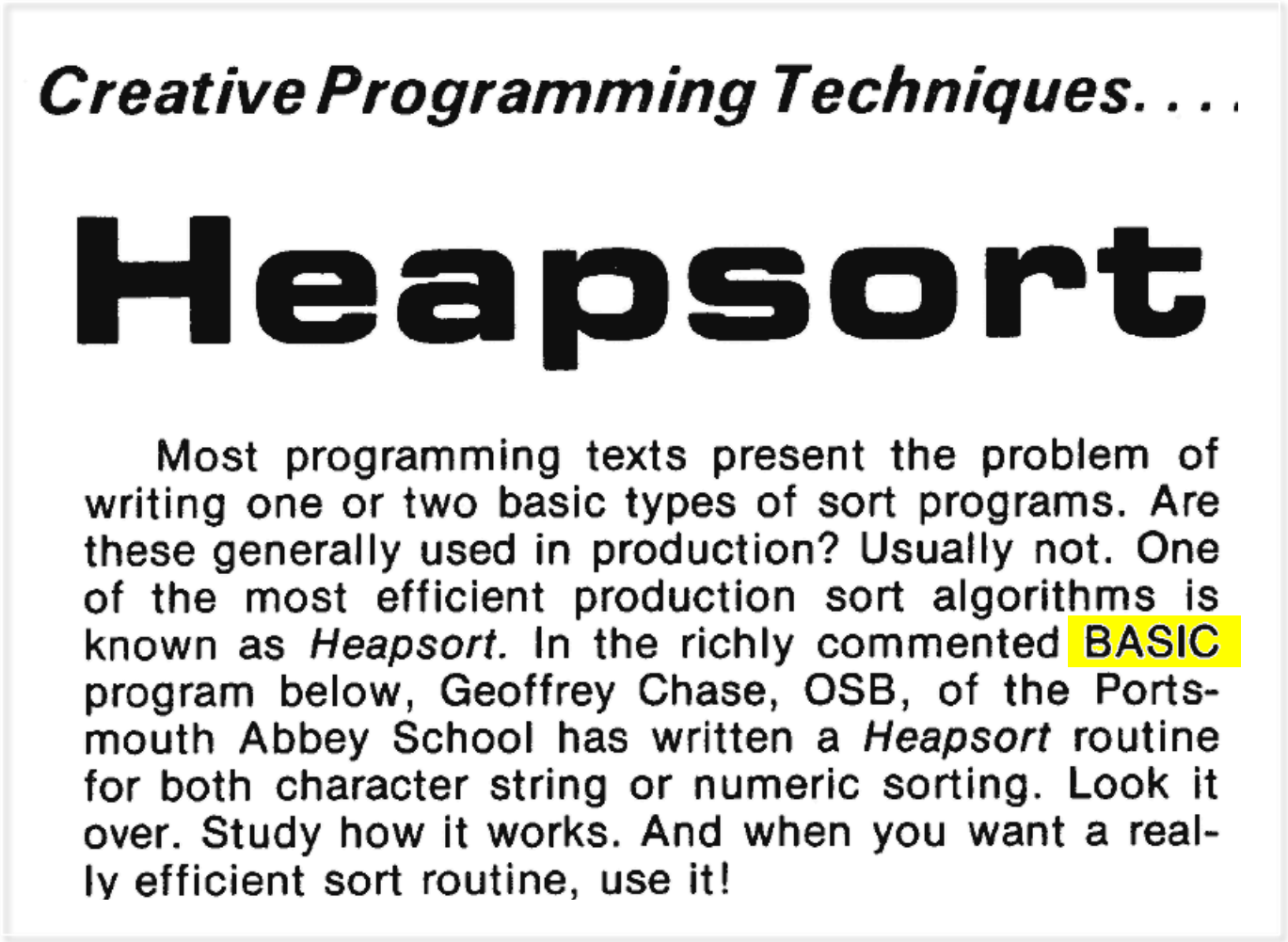
Zwei Schnappschüsse aus Phase 2





Man kann die erste Phase noch beschleunigen: Statt n mal insert anzuwenden, wendet man auf das initiale unsortierte array eine Operation „buildheap" an, die in O(n) einen Heap erzeugt: Dazu wendet man rekursiv buildheap auf den linken und den rechten Unterbaum der Wurzel an und lässt sodann die Wurzel (wie bei get_min) durch paarweises Vertauschen so weit wie möglich hinabsinken. Oder bottom-up: Die Blätter sind bereits heaps; betrachte Knoten über den Blättern bzw. über linken und rechten Teilheaps und „repariere"...

Heapsort im Jahre 1974
(Aus: Creative Computing Magazine)



Creative Programming Techniques. . . .

Heapsort

Most programming texts present the problem of writing one or two basic types of sort programs. Are these generally used in production? Usually not. One of the most efficient production sort algorithms is known as *Heapsort.* In the richly commented BASIC program below, Geoffrey Chase, OSB, of the Portsmouth Abbey School has written a *Heapsort* routine for both character string or numeric sorting. Look it over. Study how it works. And when you want a really efficient sort routine, use it!

Heapsort wurde 1964 entdeckt

Die folgenden Slides stellen lediglich eine historische Kuriosität dar.

Interessierte mögen aber Spass am „reverse engineering" des Programms haben und nebenbei einen Eindruck der seinerzeit populären Programmiersprache „BASIC" bekommen.

Die Programmiersprache BASIC wurde 1964 entwickelt: Einfach, interpretiert, interaktiv; passend zum Heimcomputer der 1970er-Jahre.

```
010 REM. KNUTH/WILLIAMS/FLOYD HEAPSORT ALGORITHM.
020 !  PAS '74
030 DIM N(150),C$(150)
040 PRINT
050 PRINT
060 PRINT "TYPE  C  FOR CHARACTER STRING SORT,"
070 PRINT "TYPE  N  FOR NUMBER SORT.";
080 INPUT W$
090 N=0                              ! START COUNT=N AT 0
100 PRINT
110 IF W$="N" THEN 480
120 IF W$<> "C" THEN 60              ! BAD REPLY
130 REM ********* CHARACTER STRING SORT ROUTINE: *********
140 GOSUB 770                        ! ASK FOR STOP CODE
150 INPUT S$                         ! GET STOP CODE
160 PRINT
170 N=N+1                            ! INPUT LOOP:
180 INPUT C$(N)
190 IF C$(N)<>S$ THEN 170
200 N=N-1                            ! END OF INPUT...
210 PRINT
220 L=INT(N/2)+1                     ! HEAPSORT PROPER:
230 N1=N                             ! PRESERVE N, USE N1
240 IF L=1 THEN 280
250 L=L-1
260 A$=C$(L)
270 GOTO 320
```

Keine Klein-buchstaben

It is practically impossible to teach good programming to students that have had a prior exposure to BASIC: as potential programmers they are mentally mutilated beyond hope of re-generation. -- Edsger Dijkstra

Keine Schleifen, sondern bedingte und unbedingte Sprünge

Das musste man Buchstabe für Buchstabe aus der Zeitschrift abtippen – es gab ja keinen Server und kein Internet, um es herunterzuladen; auch keine CDs oder Floppys!

```
280 A$=C$(N1)
290 C$(N1)=C$(1)                    ! MOVE TOP OF HEAP TO END
300 N1=N1-1                         ! HEAP IS ONE SMALLER NOW
310 IF N1=1 THEN 430                ! ONLY ONE LEFT? THEN WE'RE DONE.
320 J=L                             ! NO, CONTINUE
330 I=J
340 J=2*J                           ! LOOK FOR "SONS" OF I
350 IF J=N1 THEN 390
360 IF J>N1 THEN 420                ! "N1" IS SIZE OF ACTIVE LIST
370 IF C$(J)>=C$(J+1) THEN 390      ! CHOOSE LARGER "SON"
380 J=J+1
390 IF A$>=C$(J) THEN 420
400 C$(I)=C$(J)
410 GOTO 330                        ! LARGER SON REPLACES PARENT
420 C$(I)=A$
425 GOTO 240                        ! END OF SORT...
430 C$(1)=A$
440 FOR I=1 TO N
450 PRINT C$(I)
455 NEXT I
460 GOTO 60
```

Unwesentliche Teile hier weggelassen

Die Logik von "Basic" ist typisch amerikanisch: ohne absolute Prinzipien, dafür pragmatisch, zuweilen unberechenbar und sehr fehlertolerant. "Basic"-Programme im Pentagon würden den Untergang bedeuten, im "Weißen Haus" entsprechen sie derzeit dem Common sense. Das Gegenstück zu "Basic" ist die logisch ungeheuer klare, geradezu sture Programmiersprache "Pascal", die der Zürcher Mathematik-Professor Niklaus Wirth mit gleichsam zwinglianischer Strenge entwickelt hat. [Der Spiegel 12/1987]

```
760 REM **********SUBROUTINE TO INPUT STOP CODE**********
770 PRINT "PLEASE INDICATE A STOP CODE--SOMETHING NOT IN YOUR"
780 PRINT "LIST, WHICH WILL ACT AS AN 'END OF LIST' SIGNAL: ";
790 RETURN
800 END
```

Hobby-BASIC-Programmierer und Heimcomputer-Nutzer der 1970er-Jahre


```
1249 BYTES FREE
READY.
? LOAD "HEAPSORT"
PRESS PLAY ON TAPE
SEARCHING
FOUND HEAPSORT
? LOAD
READY.
? LIST 280
280 A$=C$(N1)
290 C$(N1)=C$(1)
300 N1=N1-1
310 IF N1=1
THEN 430
320 J=L
330 I=J
340 J=2*J
350 IF J=N1 THEN 390
360 IF J>N1 THEN 420
370 IF C$(J)>=C$(J+1) THEN
380 J=J+1
390 IF A$>=C$(J) THEN 420
400 C$(I)=C$(J)
...
```

Das dauerte einige Minuten

& machte nervige Piepstöne

Oder direkt **RUN**: BASIC ist eine interpretierte Sprache; kein Compilieren nötig

Wie konnte man damals nur (über)leben? Sind wir froh, im Hier und Heute zu leben!

Magnetband-Kassettenrekorder als Speichermedium

Die meisten BASIC-Interpreter stammten von Microsoft. BASIC war Microsofts erstes und in den frühen Jahren wichtigstes Produkt, mehrere Jahre bevor mit MS-DOS das erste Betriebssystem der Firma auf den Markt kam.

Laden von Programmen

1. Verbinden Sie Ihren COMMODORE-Computer und die COMMODORE-Datassette miteinander.
2. Legen Sie die Programm-Cassette in die Datassette ein, die erste Seite nach oben.
3. Drücken Sie die Taste REWIND, um an den Bandanfang zu spulen.
4. Wenn das Band anhält, auf STOP-Taste drücken.
5. Geben Sie über die COMMODORE-Computer-Tastatur das Wort LOAD ein und drücken Sie auf die RETURN-Taste.
6. Auf dem Bildschirm erscheint PRESS PLAY ON TAPE. Drücken Sie bei der Datassette auf die Taste PLAY.
7. Der Bildschirm wird gelöscht, während das Programm gesucht wird.
8. Nach einigen Sekunden wird der COMMODORE-Computer auf dem Bildschirm melden: FOUND (dahinter der Programmname).
9. Der Bildschirm wird automatisch gelöscht und das Programm wird geladen.
10. Nachdem das Programm geladen ist, gibt Ihnen der COMMODORE-Computer auf dem Bildschirm die Meldung READY.
11. Um das Programm laufen zu lassen, tippen Sie nun über die Tastatur einfach RUN ein und drücken auf die RETURN-Taste. Das Programm startet.
12. Sollte es nicht funktionieren, dann noch einmal bei Punkt 3 starten!

(Es funktionierte mit dieser Tonhöhenkodierung eher selten, meist gab es einen LOAD ERROR)

```
### COMMODORE BASIC ###
 7167 BYTES FREE

READY.
LOA

?SYNTAX ERROR
READY.
LOAD

PRESS PLAY ON TAPE #1
OK

SEARCHING
FOUND INVADER
?LOAD ERROR
READY.
```

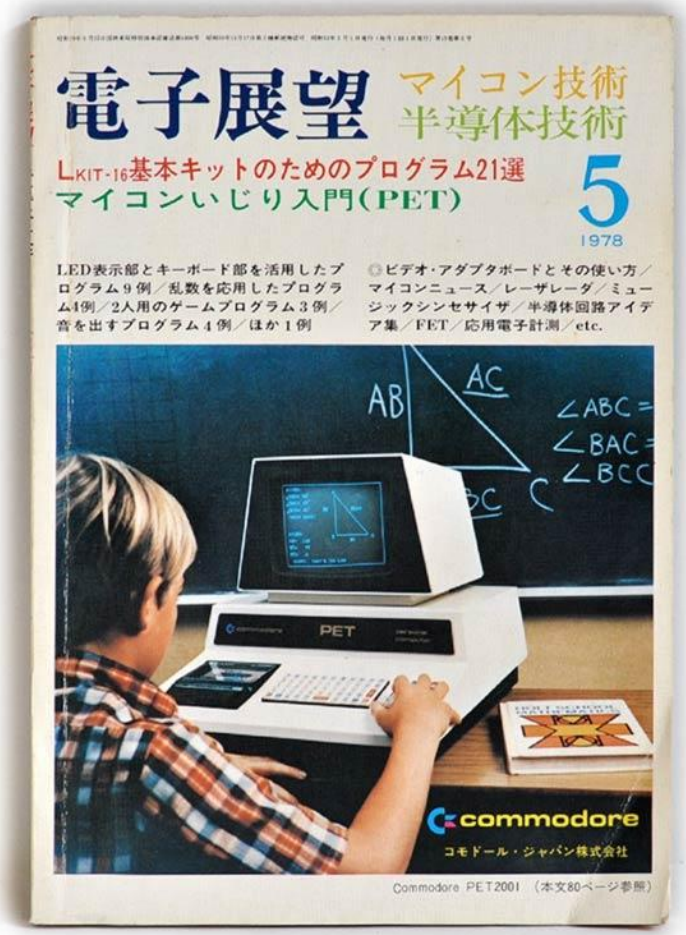
It's hard for kids to believe now that we often had to wait 15 minutes for a game to load. So, what did we do while we waited for games to load? Wrestle! Many a loading time was passed wrestling with my school friends on sofas, beds and carpets.
-- commodorekid.blogspot.com/2017/01/

1977 Der erste direkt nutzbare Homecomputer

▸ Eingebautes Basic im ROM; ▸ Hauptspeicher erst 4kB, dann 8kB; ▸ CPU: MOS 6502, 1 MHz Taktfrequenz; ▸ Monochromer Röhren-Monitor (40 × 25 Zeichen); ▸ Blockgraphik mit 320 × 200 Pixel.

Der Commodore PET 2001 (Personal Electronic Transactor) wurde ab Juni 1977 für 795 US-Dollar vertrieben, in Europa kam eine Variante unter der Bezeichnung „CBM 3000" im Frühjahr 1978 auf den Markt.

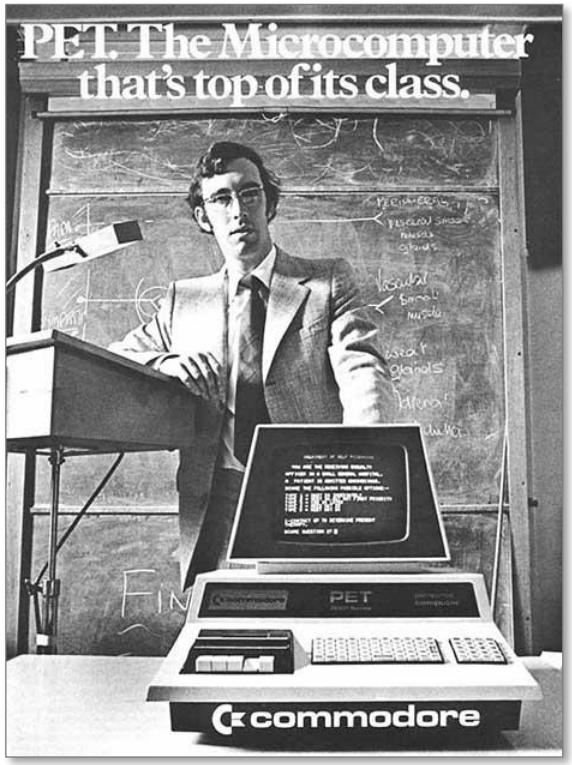
„The PET was a revelation as all previous home computers were little more than circuit boards that could only be understood by hard core enthusiasts. At the time Commodore manufactured office equipment like filing cabinets but its biggest business was in calculators, so it is no surprise that the original production Commodore PET 2001's had a sheet metal chassis and calculator style keys dubbed 'chiclet keyboards'. When Commodore expanded to Europe in 1978 Jack Tramiel doubled the price for the same machine but the only physical change was a 220 watt power supply. The rebranded 3000 series were highly successful in the European markets at the higher prices. The Basic Operating System was written by Bill Gates and Paul Allen from their fledgling 'Micro-Soft' Corporation. Commodore Basic was the only unlimited software license ever granted by Microsoft to any company for all products regardless of the number of copies used." [www.commodore.ca/commodore-products/commodore-pet-the-worlds-first-personal-computer/]



▸ „PET 2001. Ein Rechner, der seinen eigenen Monitor und dazu Kassettenlaufwerk, Tastatur und Netzteil mitbrachte. All das in einem stabilen Stahlgehäuse. Bereit zum Aufstellen und Benutzen."

„PET 2001"? Die Zahl „2000" hatte in den Jahren und Jahrzehnten vor der Jahrtausendwende einen futuristischen und fast magischen Klang – das Jahr 2000! Die „2000" tauchte sogar in Namen von Produkten auf, um diese besonders modern erscheinen zu lassen.

Aber Zeit vergeht, und irgendwann war „2000" nicht mehr futuristisch sondern drohte antik zu werden. Die Firma „Gateway 2000", in den 1990er-Jahren ein grosser amerikanischer PC-Vermarkter, änderte z.B. im April 1998, gerade noch rechtzeitig vor dem Millennium, ihren Namen, indem sie schlicht zu „Gateway" umfirmierte. "The company is yanking `2000´ from its name, loath to appear out-of-date by letting the millennium creep up on it", schrieb das Wall Street Journal am 24. April 1998.



In den 1970er-Jahren lag das Millennium noch unvorstellbar weit in der Zukunft, und „2001" übertraf „2000" sogar noch in seinem futuristischen Beiklang. Der damals sehr angesagt Science-Fiction-Film „2001: Odyssee im Weltraum" von Stanley Kubrick aus dem Jahr 1968 verlieh der Zahl 2001 noch zusätzlich eine futuristisch-mystische Aura. Mit einem „PET 2001" konnte man sich somit einen Hauch von Zukunft und Science-Fiction ins Haus holen, und vielleicht ja ganz konkret auch ein Stückchen vom fantastischen HAL-Computer im Film, der Fähigkeiten hat, die man sich selbst gar nicht vorstellen konnte, bevor man den Film sah. Und 2001 als Verfallsdatum? In über 20 Jahren hätte der PET wohl ausgedient und wäre zurecht eine echte Antiquität…

Das Akronym PET (Personal Electronic Transactor) andererseits kann im Englischen als „Haustier" oder „Liebling" verstanden werden – sympathisch und nett würde der PET 2001 also auch noch sein.

1980 findet sich der PET 2001 und sein CBM-Nachfolgemodell im Quelle-Bestellkatalog. Der Name musste geändert werden, da die Firma Philips die Abkürzung PET für „Programm-Entwicklungs-Terminal“ nutzte. Es gab nun ein Floppy-Laufwerk anstelle des Kassettenrekorders. Zusammen mit Computer und Drucker kostete das System fast 6000 DM – das war der halbe Preis eines neuen VW Golf.

Jetzt bei Quelle: ein komplettes Computer-System!

. . . zu einem bisher nicht für möglich gehaltenen Preis. Damit können jetzt die Vorteile des Computers auch in kleinen Unternehmen und im privaten Bereich voll genutzt werden.

⑤ Nur noch **1898.-** Preissenkung!

③ Tabelliererpapier **29.90**

⑥ Buch **24.-**

④ Speichereinheit ab **1350.-**

② Matrixdrucker **1850.-**

① Zentraleinheit **2649.-**

① ✻ **Tisch-Computer CBM 3016–3 eine Zentraleinheit mit vielen Vorzügen**

- **Frei verfügbarer 16 kByte Schreib-Lese-Speicher (RAM).**
- **Schreibmaschinen-Tastatur (ASCII Standard) mit Sonderzeichen. Umschaltung auf Groß- und Kleinschreibung mittels BASIC-Befehl.**
- **Grüner 23-cm-Bildschirm mit 1000 Zeichen (25 Zeilen zu je 40 Zeichen). Buchstaben, Ziffern und grafische Zeichen können in negativer und positiver Schrift dargestellt werden.**
- **13 kByte Festwertspeicher (ROM) mit 8 kByte BASIC-Interpreter, 4 kByte Betriebssystem, 1 kByte Testroutine.**
- **Anschluß für Kassettenrekorder sowie Interface IEEE 488 zum Anschluß von bis zu 12 Peripheriegeräten, 8 Bit breiter User Port.**

Programmierbar in BASIC. Maschinensprache ist möglich (Mikroprozessor 6502). Jeweils 960 Einfach-, Ganzzahl- und Stringvariable. 12 mathematische Funktionen. Maße ca. 42 × 47 × 36 cm. Gewicht ca. 20 kg.

Komplett mit Bedienungshandbuch

Bestell-Nr. 023.975 DM **2649.-**

① ✻ **Tisch-Computer CBM 3016–2.** Gleiches Gerät wie CBM 3016–3 – jedoch für Groß- und Kleinschreibung. Umschaltung auf graphische Sonderzeichen mittels BASIC-Befehl.

Bestell-Nr. 023.974 DM **2649.-**

② ✻ **Matrixdrucker CBM 3022.** Mit diesem Drucker können alle Zeichen auf Normalpapier ausgedruckt werden: Groß- und Kleinbuchstaben, Ziffern und Sonderzeichen. Sperrschrift ist möglich, ebenso Negativschrift. Durch die exakte Traktorführung können Formulare und Rechnungen bedruckt werden, ebenso wie Selbstklebeetiketten von der Endlosrolle. Druckbreite, Dezimalpunkt, Wegfall von Nullen vor oder hinter dem Komma, links- oder rechtsbündiges Schreiben, Anzahl der Zeilen pro Seite und Zeilenabstand 1/6 oder 1/8 Zoll lassen sich programmieren. Normalpapier mit Lochrand (Breite von 8″ bis 10″ variabel) 4 Kopien sind möglich. Druckgeschwindigkeit max. 150 Zeichen/Sec; 80 Zeichen pro Zeile. **Serienmäßig ausgestattet mit IEEE 488 Interface.**

Bestell-Nr. 023.976 DM **1850.-**

③ ✻ **Endlostabelliererpapier.** 1000 Blatt mit Lesehilfslinien, 240 mm breit, 8″ lang, abtrennbar.

Bestell-Nr. 008.111 DM **29.90**

④ ✻ **Speichereinheit CBM 2032** mit zwei Magnetplattenlaufwerken (Doppel-Floppy Disk) für Standard-Disketten (Speicherplatte 5,25″). Ihre hohe Übertragungsgeschwindigkeit macht sie als Eingabegerät für Programme und Daten unentbehrlich. Die Floppy Disk 2032 hat ein eigenes Betriebssystem und ist somit unabhängig von der Zentraleinheit organisiert. Als sogenanntes „intelligentes“ Gerät beansprucht sie keinerlei Speicherkapazität in der Zentraleinheit. Die Speicherkapazität beträgt 2 × 160 kByte. **Lieferbar ab September.**

Bestell-Nr. 023.955 DM **1995.-**

✻ **Diese Artikel werden separat ausgeliefert. Näheres auf Seite 955.**

④ ✻ **Single Floppy Disk CBM 2031** – gleiches Gerät wie vor, jedoch mit nur einem Laufwerk. Kapazität 160 kByte. **Lieferb. ab Sept**

Bestell-Nr. 023.954 DM **1350.**

✻ **Anschlußkabel** für ein Peripheriegerät an Zentraleinheit.

Bestell-Nr. 023.956 DM **115.**

✻ **Verbindungskabel** zwischen zwei Peripheriegeräten, Drucker ↔ Floppy Disk.

Bestell-Nr. 023.957 DM **139.**

✻ **10 Disketten,** 5,25″ für CBM 2031 und 2032

Bestell-Nr. 023.965 DM **119.**

Ohne Bild ✻ **C 2 N – Kassettenrekorder** zum Speichern von Programmen und Daten. **Serienmäßig mit Anschlußkabel an Zentraleinheit.**

Bestell-Nr. 023.946 DM **295.**

⑤ ✻ **CBM 2001 – ein Basic-Tischcomputer mit alphanumerischer Tastatur inkl. Sonderzeichen, Kassettenrekorder sowie frei verfügbarem 8 kByte Schreib-Lese-Speicher (RAM).** Anschlüsse und Ausstattung wie CBM 3016. **Komplett mit deutsch. Bedienungshandbuch**

Bestell-Nr. 004.312 nur noch DM **1898.**

✻ **BASIC-Lerncassette** – Einführung in die Technik des Programmierens mit BASIC.

Bestell-Nr. 004.334 DM **39.**

✻ **Schach-Spielcassette.** 8 Schwierigkeitsgrade. Darstellung von Schachbrettfiguren und Uhr

Bestell-Nr. 008.243 DM **79.**

Ohne Bild ✻ **Matrixdrucker TX 80 zu CBM 2001** – mit Traktorführung. Mit diesem Drucker kann der volle ASCII Zeichensatz, Groß- und Kleinschreibung und die grafischen Zeichen des Heimcomputers 2001 auf Normalpapier ausgedruckt werden. Druckgeschwindigkeit 125 Zeichen/Sec. bei 80 Stellen pro Zeile. **Serienmäßig ausgestattet mit IEEE 488 Interface und Kabel zum Anschluß an CBM 2001 und somit sofort betriebsbereit.**

Bestell-Nr. 038.071 nur noch DM **1798.**

⑥ **Programmieren von Heimcomputern in BASIC.** Mit zahlreichen Beispielen und vollständigen Programmen. 140 Seiten.

Bestell-Nr. 611.754 DM **24.**

1977

Home computers can change your life-style

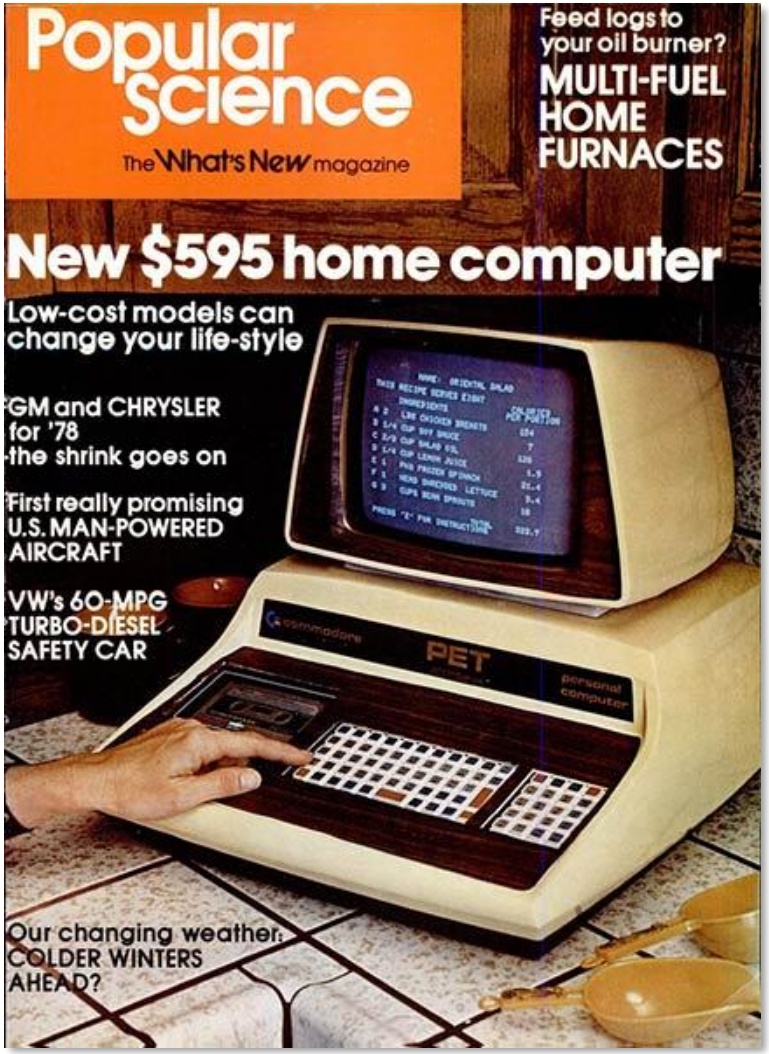
William J. Hawkins. In: Popular Science 211(4), 1977:

Until now, to operate a home computer you had to be part electronic engineer, part programmer, part handyman, and part lucky. […] Most hobby units available have been pretty complicated and time-consuming. Success in building and operating them has been reserved for the devoted electronic enthusiast.

But that's all changing: A new generation of home computers is beginning to emerge. With these, you take 'em home, plug 'em in, and begin using them. […] Now – or very soon – everyone can have at his command affordable computing power, power that until recently has been available only to corporations and other big-money institutions. It's bound to have a major impact on the way we all live, in ways we can now only dimly imagine. […]

Computer shops and clubs are springing up across the country. New magazines, books, and newsletters dealing with the subject are appearing monthly. […]

Sophistication and ease of operation are headed up while prices are headed down. Units are not designed just for the hobbyists, but for almost anyone who can put in a plug. The Commodore PET unit is a prebuilt, all-in-one device that can be programmed by anyone. You can become an expert in a couple of hours. […]



If you're worried about programming, don't – you can be an expert in just a few hours learning the standard BASIC language that these machines will use. BASIC is simply that: a basic, simple way to communicate with a computer. […]

Okay, the prices are dropping. the computers are simpler to use, and they're easily available. But what will you do with them around the house?

On a diet? Tell the computer and it will plan your menu for each meal, forecast your caloric intake, and predict your weight loss for the next month.

Need a loan (perhaps to pay for the computer)? It can compute when and where to get it, how and when to pay it back to cost you as little as possible.

Budgets, income taxes, mass storage (perhaps of frequently called phone numbers) for instant recall, "canned" programs to tutor your children at home, appointment reminder, or games such as chess – a computer can do those and many other jobs.

But the ultimate impact these home computers will have on our lives will, I believe, be far greater than just playing games or balancing our check-books. Major inventions have always changed our society in ways that could not be predicted.

1978

Let the computer do it!

„Homecomputer – wie der Name schon sagt, für Kochrezepte und Kalender gedacht." – Britta Schinzel

Madison Wisconsin State Journal, September 18, 1978:

If your schedule is too hectic to worry with the details of life such as the balancing of your checking account, or if one more game of checkers with the kids will drive you crazy, for a small investment you can now buy a computer to do the things you don't want to do.

For around $800, any family can buy a home computer […] According to Huron Smith, owner of the Madison Computer Store, […] the computer will do practically anything you tell it to do, but it never does anything on its own. "They're absolutely stupid," she said. "They have no smarts of their own. They have to be programmed." A home computer can be programmed with ordinary words so a buyer doesn't need any special skills. The →

LET PERSONAL COMPUTERS DO IT.

Do what? Just about anything. Your imagination is the limit. And, nobody knows this better than ComputerLand.

So first, let's take a look at some of the many things an **Apple** computer can do, and then the best way to buy your own personal computer to do it.

Home Environmental and Financial Control. Your **Apple** personal computer can make you the master of the household environment. It can control your heating and air conditioning. Run a security system. Automate the garden's watering. Control a solar energy unit, and more.

An **Apple** personal computer can also be your private financial counselor. It can examine planned investments. Monitor the household budget. Compute your taxes. There's even an optional hook-up that can connect it via telephone, to what's happening on Wall Street.

An Office Computer in the Den. There are a good many business problems a computer can solve. And, ComputerLand can show you how. So, if you're not as conversant with the technology as you'd like to be, starting with your own personal computer at home is a good way to get involved.

If you already use a computer on your job, having one at home solves the problem of waiting in line to use the big one at work. Develop your scientific, engineering and business programming ideas in the evening.

Then dazzle them on the job in the morning.

© ComputerLand Corp., 1978

The Ultimate Educational Tool. Perhaps the best thing about a personal computer, however, is not what it does, but what it is. And that is an exciting and inventive tool for learning. One that's being used more and more in the classroom. So having an **Apple** computer in the home is like having your own private tutor.

The whole family can start designing simple games, just for the fun of it. Before long, even a young child will progress to intricate programs, with an intuitive insight into computer technology.

You'll Find It, At ComputerLand. Everything you've been looking for. A computer store that has all the equipment you've read about and then some, plus demo areas for you to try them out. A professional staff to answer all your questions. And, a fully equipped service center that will provide whatever technical assistance you may need.

No matter where you're located, there's a ComputerLand near you. Visit us today. We'll make it easy for you to own a personal computer to do it all.

ComputerLand®

We Know Small Computers.

Over 100 Stores Worldwide
For Your Nearest Computerland Stores Call Toll Free:
(800) 227-1617, Extension 118
California, (800) 772-3545, Extension 118

Franchise Opportunities Available

1978

Let the computer do it! (2)

domestic variety of computers are programmed with BASIC language, which in computer talk is simple English. Len Lindsay, a salesman at the Computer Store, said, "The name implies it's very easy; it took me three days to learn it."

Since May, more than 40 Madison families have turned over their tedious memory and budget chores to their computers. More and more Madison families are programming machines to keep track of Aunt Minnie's birthday and the grandparents' anniversary. Keeping up with the addresses of mobile friends is now the computer's job. And it's no problem to plan for four dinner guests, she can relegate that duty to the machine. The computers are unbeatable as managers of household finances. […]

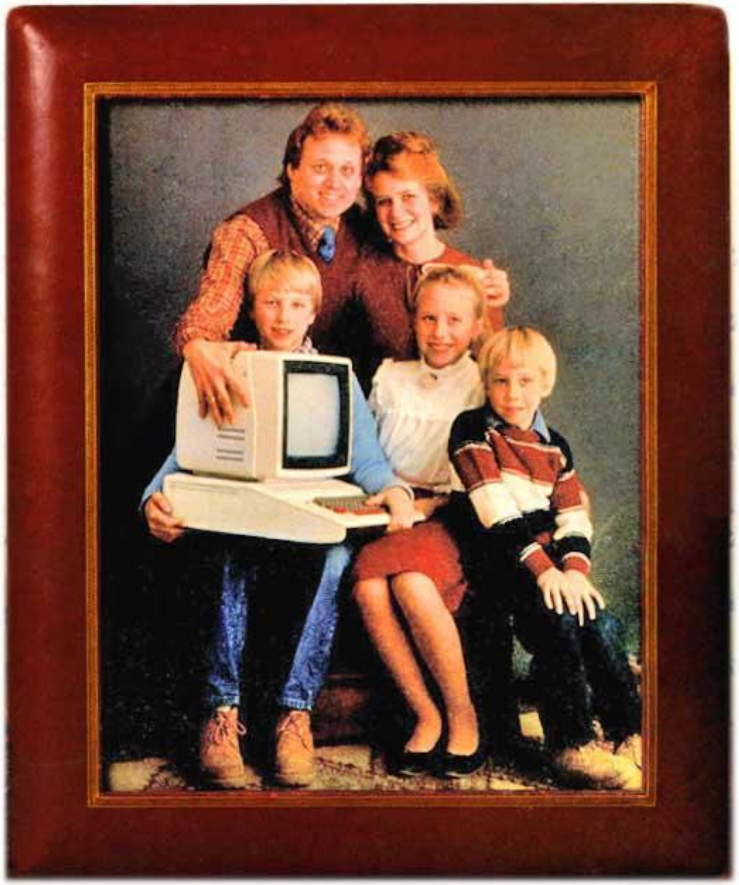
→

1978

Let the computer do it! (3)

The computer is also an able educator. There are programs to teach children to tell time, and programs for math games, decimal division, history quizzes and to tutor morse code. Lindsay said computers are perfect for kids who need a high level of motivation because computers can be programmed to offer any type of reward from words of praise flashed on the screen to shooting stars. […]

Once you get your computer home, you can add accessories. According to Lindsay, a lot of people in Madison are adding sound. It only takes two wires and costs about $10. Anyone who comes into the store and tinkers with the machines for half an hour wants to have one, Lindsay said. "It's a time saver," he said. "It'll do all the hard things for you plus, as a side benefit, you can sit and play games for free."



Die meisten Käufer von damals hatten weniger einen konkreten Nutzen im Sinn – auch wenn sie den gern vorschoben –, sondern sie wollten einfach auch einen Computer haben, den Inbegriff des Fortschritts und der Moderne. -- Wolf Lotte

1978

Computers in the homes? Wave of the future is now!

"Computers are finding their way into the home"

It's a weekday evening, sometime in the near future. This Bucks-Mont family has just finished dinner and is settling into its normal weeknight routine. The children go to the VDTs in their room to work on their homework. […] Mom putting the dishes in the dishwasher and programs the central processing unit to do the dishes and, while she's at it, gives the unit new climate control instructions. Meanwhile, Dad is on his way to the store to buy that much-needed additional 2K memory he and the wife have been meaning to get. […]

What it is is the family of the future — completely equipped with its own personal computer. The world that many thought could only exist in a dream or a science fiction […] is finally here.

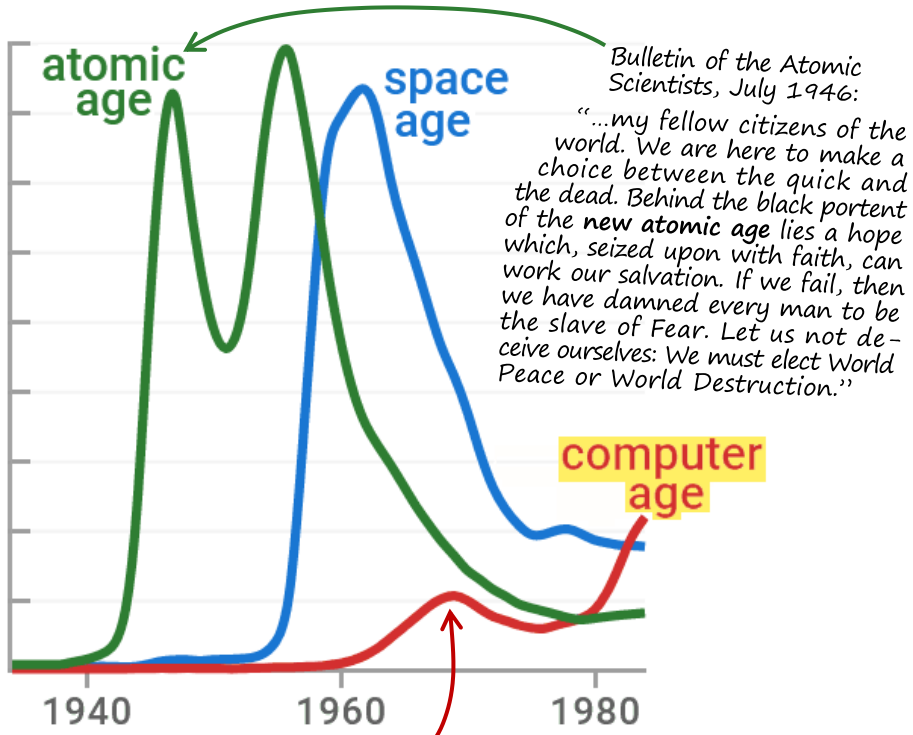
Computers […] are finding their way into the home. The breakthrough came last year when several firms started marketing home computers — usually composed of a typewriter panel, a television screen and a cassette recorder. The machines now being marketed cost no more than a high quality stereo, and require little more than reading an instruction manual to learn how to operate. The home models presently available can store recipes, do math problems, and even record telephone calls when a person is not home. […] Someday the machines will be doing everything from the wash to income tax returns.

Donald French, merchandising manager for the Tandy Corporation, said his company's market research shows a "very good" potential for home computer sales. […] And many people involved in the industry are predicting computers will become the ultimate home life status symbol. […] French said he and others in the industry are certain the home computer will revolutionize home life, making still more time available for leisure and other activities.

Doylestown Intelligencer, March 18, 1978

1982 The Computer Age Is Here

Das Weekly-Reader-Magazin wurde in den USA über die Schulen vertrieben, es erschien freitags. Die „Edition 5" (hier die Ausgabe vom 8. Januar 1982) war für Fünft- und Sechstklässler bestimmt.



Die erste Computer-Age-Welle ist den Grossrechnern geschuldet; ab Ende der 1970er-Jahre erreichen Computer in der Form von Heimcomputern (und später PCs) auch Schülerinnen und Schüler.

1982

Ein Homecomputer für die lieben Kinder

Wie kann man einen eher teuren Homecomputer verkaufen? Die Firma Texas Instruments, welche damals Vielen aus eigener Gymnasialzeit oder Studium aufgrund der hochwertigen Taschenrechner ein Begriff war, wandte sich mit der Werbung in bildungsaffinen Medien direkt an die Eltern – schliesslich waren sie es, die einen Homecomputer für die Familie und damit für die Kinder kaufen und bezahlen sollten. Allein die Konsole des TI-99/4A kostete im Jahr 1982 etwa 1500 DM, was mehr als ein halber durchschnittlicher Monatslohn war, dazu brauchte man noch einen Fernseher als Ausgabemedium sowie Software-Module.

Aber was tut man nicht alles für seine Kinder und deren Bildung! Verführerisch klingt da die Werbung für Papi, der schon lange mit einem Homecomputer für sich liebäugelte: „*Nehmen Sie das Modul ‚Rechenkünstler', stecken Sie es in den TI-99/4A, und schon beginnt ein äußerst lehrreicher Dialog zwischen Ihrem Kind und dem Computer. Und nach dem Rechnen ist Musik genau die richtige Entspannung. Mit dem Modul ‚Music Maker' lernt Ihr Kind ganz schnell den Unterschied zwischen Viertel- und Achtelnoten und Dur und Moll.*"

1983 Ein Homecomputer zum Spielen und mehr

Der Atari 600XL kam im Herbst 1983 auf den Markt, er kostete 549,- DM in Deutschland (also nur etwa ein Drittel des TI-99/4A). Atari warb einerseits mit dem spielerischen Lernen von Programmieren in BASIC und „nützlichen" Anwendungen für die ganze Familie:

„Endlich vergißt der Vater den Hochzeitstag nicht mehr, und er weiß auch noch sofort, wieviel Geld er für den Diamantring übrig hat. Die Mutter möchte wissen, welche Zutaten sie für einen Vanillerostbraten für sechs Personen braucht und ob für Peter in diesem Monat noch neue Schuhe drin sind. Peter muß noch ein paar Übungen für die Schule machen. Und möchte anschließend sein Schallplattenarchiv ordnen."

Andererseits wurde mit der „Menge lustiger und spannender Spiele" geworben, zu denen beliebte Klassiker wie Donkey Kong und PAC-MAN gehörten „mit hervorragenden Grafiken und unglaublichem Sound". Der Atari 600XL wird jedoch nur ca. ein Jahr lang produziert.

1983

Switch it on!

http://commodorekid.blogspot.com/2017/01/do-you-remember-first-time.html (gekürzter Auszug):

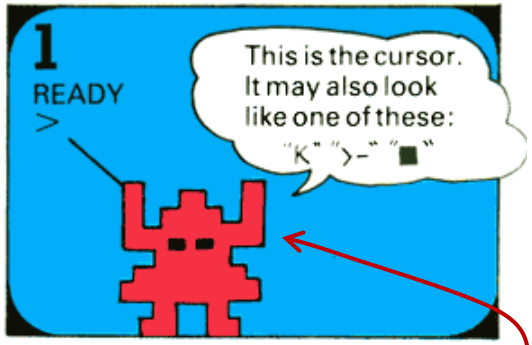
It was a windy autumnal day when Dad proudly strode through the front door with it under his arm. It came in a long white oblong polystyrene box which had a cardboard sleeve that slid over it. We carefully unsheathed it and opened the polystyrene box. Inside was the computer itself – encased in cream coloured plastic with dark brown keys and four fat orange function keys. The box also housed a heavy fudge coloured wedge shaped thing, a small silver tin box, a couple of black cables and a spiral bound manual. We laid out all these pieces on the coffee table in the front room and gazed upon them in wonder as if they were the parts of a space ship that had crash landed in the back yard.

The next task was easy. I took that duty upon myself and pulled out the Ferguson TV from the corner of the front room, removed the TV aerial and pushed the computer cable into the TV's aerial socket. Done. This was it. The big moment. Mum was called in from the kitchen and Our Susan was summoned from her bedroom to witness the momentous occasion. "Ruth! Ruth!" Shouted Dad to Mum. "Susan! Susan! We're switching the computer on!" I yelled upstairs. Soon the family was gathered around the computer and the excitement was building. I could hardly contain myself.

"Well, go on Martin, switch it on," said Dad. After a quick check that all the cables were secure, I flicked the switch on the side of the VIC and a red LED light blinked into life. There was a sudden increase in tension in the room. I clambered around the coffee table, careful not to trip over the 2-meter-long wire that connected the computer and the PSU to the power socket, and switched on the TV. This is it! I thought. It was like lighting a firework. I had lit the fuse and now it was time to stand back and watch the splendour and magic unfold before our eyes. Unfortunately, like so many fireworks, we were rather disappointed when nothing happened. "What's wrong with it?" asked Mum. "I don't know." said Dad. "Let me have a look," I said. I checked all the cables but everything looked fine. "I'll be upstairs if you need me," said Our Susan rolling her teenage eyes and disappeared back to her bedroom and Duran Duran. →

1983 Switch it on! (2)

I picked up the manual again, had another read through the first few pages and spotted the error of our ways. The TV had to be tuned to the computer's signal. I pressed the 6 button and flipped open the little secret door on the front of the telly that hid the tiny wheels that you used to tune the stations in. After a bit of wheeling – first all the way down and then quite a long way back up – a ghostly image eventually slid into view and as I slowly tweaked the wheel to get as sharp a picture as possible this appeared on the screen:




```
3883 BYTES FREE
READY.
>
```

Mum and Susan were summoned once again and this time both were suitably impressed. Everyone had a go at typing their name in and we all wondered in disbelief at how we could make our own name appear on TV. After a shaky start, the VIC was a hit and a few minutes later we stared at it with further astonishment when Dad learned this impressive piece of early BASIC programming from the manual:

So eine grosspixelige Gestalt verstand man damals direkt als Ikone des Computerzeitalters; auch wenn der Bresenham-Algorithmus sein Bestes tat, kontinuierliche Kurven glattmöglichst zu rastern!

```
10 PRINT "HELLO"
20 GOTO 10
RUN

HELLO
HELLO
HELLO
HELLO
HELLO
HELLO
```

Now that blew our minds.

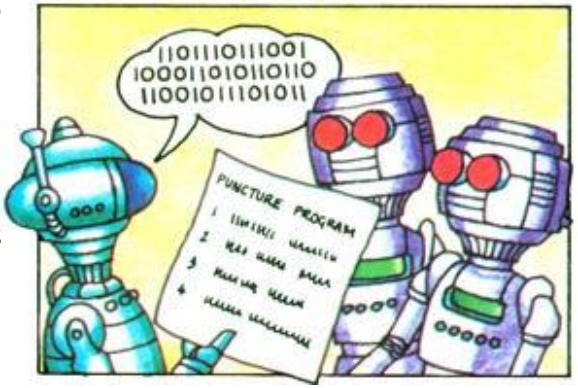
If all of the above seems a bit unlikely, you have to remember that this was three decades ago. Before computers existed, you could only watch the television – you had no way of affecting what was on your TV screen, you just switched it on or off and watched whatever was being broadcast at the time. You were passive. But this was different. Now you could make whatever you wanted appear on your TV screen. You were active. Empowered. It was a revelation. A revolution.

1980er-Jahre

Today, Computers Should Interest Everybody

Der britische Heimcomputer ZX-80 der Firma Sinclair erschien im Januar 1980, das Nachfolgemodell ZX-81 im März 1981. Er war als Massenware für Einsteiger konzipiert. Entsprechend dem *UK General Household Survey 1984* besassen 1984 9% aller britischen Haushalte einen Heimcomputer. James Sumner beschrieb 2012 das Aufkommen der britischen Heimcomputer Anfang der 1980er-Jahre ['*Today, Computers Should Interest Everybody'. The Meanings of Microcomputers.* Zeithistorische Forschungen 9.2, 307-315]; ein Auszug:

"What, however, did the new users expect to use the machines for? Early publicity employed notorious artistic licence in describing the machines' versatility: the ZX-80, supposedly, could do 'quite literally anything from playing chess to running a power station'. Such claims soon disappeared, not only because they were implausible, but because mass-market customers did not run power stations. The marketing cliché which arose instead […] was firmly domestic: microcomputers would store recipes, manage the household accounts, provide entertainment, and educate the children.

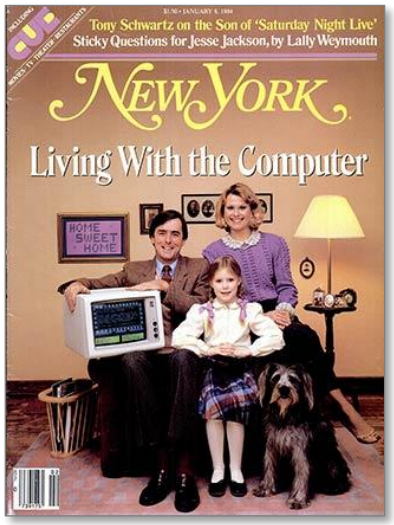


Arguably, this presentation was equally far from reality. The limitations of most of the early machines, chiefly in rapid data storage and retrieval, made them highly inconvenient for many of the uses discussed. Yet the first domestic computers did not have to be usable – in the sense accepted today – to be saleable. Leslie Haddon, author of the first significant sociological study about the British case, usefully considers the ZX-81 and similar machines as 'self-referential' computers, valuable chiefly as tools to explore the concept of a computer itself, and in particular its ability to be programmed with new, possibly user-generated instructions. Many users, indeed, bought computers (often at considerable sacrifice) without knowing why, beyond a general sense that they would be 'really important' in negotiating the future. One convert neatly described the ZX-81 as the 'motorbike of computers': unconventional, for some purposes impractical, but the fastest cheap means of getting to unknown places."

1984

Living With the Computer

Gekürzter Auszug aus dem Live-Style-Magazin „New York“ vom 9. Januar 1984:



22

Living With the Computer

By Patricia Morrisroe

A year ago, Paul Somerson bought an I.B.M. P.C. Now he spends nineteen hours a day at his computer, and he can't wait until Terry, his girlfriend, gets her own. "Then she could sit in one corner of the room," he says, "and I could sit in the other. Gosh, that would be perfect." Paul and Terry are part of America's new computer culture. Patricia Morrisroe joined the computer people to glimpse their (and our) new world.

"I'm home," shouts Donald David. Five minutes later, his wife, Evalyn, and two children, Matthew, nine, and Alyssa, four, emerge from their rooms. "Dad," says Matthew, "we were working on our computers."

Matthew and Donald talk about RAM's and ROM's. … While Matthew is proud of his father's facility with the I.B.M., Donald is even prouder of Matthew's expertise with his new Commodore 64. "When Matthew outgrew the VIC-20," David says, "we upgraded him to a Commodore, and then we bought him a disc drive. Depriving Matthew of the computer is one of the worst punishments we can inflict." He also believes the computer functions as an important learning device. "My son is very bright but not well disciplined," he explains. "Right now, he's upstairs working on a math-drill program. While he would hate doing that for school, he thinks it's fun on the computer. I'm convinced kids relate to anything that looks like a television." Donald has taken a similar educational approach with Alyssa. "I started to worry when she wasn't showing any interest in learning her letters and numbers," he says. "So I bought her a Texas Instruments. She loves it and calls it her 'puter.' Now she is learning." Some parents are afraid computers will make their child less sociable, but Donald believes it has actually helped Alyssa become more outgoing.

Because she had to manage her time more efficiently, Evalyn learned to use the computer. "At first I was scared to death," she says. "I have a very bad history with machines. I look at them and they break." Once Evalyn learned how easy it was to run the word-processing program, she discovered the computer considerably changed the economics of her practice.

1984

Living With the Computer (2)

"I'm a terrible typist," Evalyn says. "I can't finish a sentence without making a mistake. There was no way I could survive in my practice. With the computer, my partner and I can do our own drafting and typing at the same time."

Evalyn has been using her husband's machine so much that Donald recently bought another I.B.M. for himself. He also bought one for the office because he doesn't want to be anywhere without it. "Now I can sit by a pool and still do my work."

"Something awful may happen," Matthew says. "My best friend might move away. Maybe we can get some equipment so we can talk to one another on the computer. Maybe in a couple of years I'll even be able to see my friend on the computer monitor, so it won't matter where he moves." While the idea of reaching out and accessing someone through a machine seems odd and perhaps a bit silly, experts say it is not a fad.

1984

Living With the Computer (3)

"With my modem, I can talk to complete strangers. It's bringing all sorts of people together like never before," says Jim Chposky, a contributing editor of List, a computer magazine.

Today, only about 2 percent of the population owns a machine. But these computer people are a prophetic fraction, young, educated, and on the rise. They are actually living with computers every day. Thus, the individuals and families portrayed in this story provide a glimpse of what the future may be like.

"The whole trend is just amazing," says Daniel Doman, a manager of Computerland. "A few years ago, the only people who bought these machines were 'hackers,' the real hobbyists. Now I get the pin-striped Wall Street crowd. Computers have not only become respectable, they've become essential. I don't know how anyone can function in the twentieth century without one."

At the Spence School, students from first grade are taught courses in microcomputing. "It's an important part of the curriculum, just like English and math," explains Christopher York, director of the computer center. York says there is a lot of parental pressure on private schools to get computers. "No one wants their child to be left behind," he says.

There was a time when Paul Somerson and his girlfriend, Terry, enjoyed doing things together. They liked to cook, go bike riding through Central Park, and spend Saturday evenings in bed with the Sunday Times crossword puzzle. Now the puzzle remains untouched, and Paul can't remember the last time he cooked dinner. These days he eats a lot of take-out Chinese food. "It's painful for me to admit this, but now that I have an I.B.M. PC, it's hard to fit Terry—or anything else—into my life."

"When I wanted to be creative, I used to cook, play the guitar, and do graphics. Now I can make the computer play music and produce fantastic pictures. That's one of the problems of this machine. It's so seductive. It practically grabs you by the shirt and says, 'Look at all this neat stuff I'm capable of doing. Spend more time with me. I'll give you anything you want.'"

"These days, you can really separate people into two groups: those who own computers and those who don't. … I can't believe there are people who still use typewriters. You have to wonder what Shakespeare would have accomplished if he'd owned a word processor," Paul says.

Homecomputer – wie es begann

1975 Auszug aus https://blog.hnf.de/der-revolutionaer-altair-8800/:

THE era of the computer in every home – a favorite topic among science-fiction writers – has arrived! It's made possible by the POPULAR ELECTRONICS/MITS Altair 8800, a full-blown computer that can hold its own against sophisticated minicomputers now on the market. And it doesn't cost several thousand dollars.

So begann der Artikel über den Altair 8800 im Januar-Heft 1975 der „Popular Electronics". Das Cover pries den Altair als ersten Minicomputer-Bausatz der Welt an. Bastler erhielten ihn für 397 Dollar. Die Gattung, in die der Altair gesteckt wurde, umfasste die kleinsten Elektronenrechner. Minicomputer bildeten die Kategorie unterhalb der Mainframes von IBM oder Control Data. Sie waren so groß wie Kühlschränke und verbreiteten sich seit den 1960er-Jahren in Firmen, Ämtern, Universitäten und Schulen. Die Minis hatten aber vierstellige Preise und standen nicht in Privathaushalten. „Popular Electronics" erweiterte die Gattung mit dem Altair nach unten. Der Altair nutzte den Intel 8080; er war seit April 1974 erhältlich. Schöpfer des Altair war der 33-jährige Ed Roberts. Seine Firma MITS – Micro Instrumentation and Telemetry Systems – saß in Albuquerque im US-Bundesstaat New Mexiko.

Ohne Zubehör war der Altair fast nutzlos; die Eingaben geschahen per Kippschalter, die Ausgaben durch LEDs. Der S-100-Bus nahm aber Karten mit Speicherchips und für den Anschluss von Peripheriegeräten auf. Der User musste dafür natürlich zahlen. Ab Juli 1975 lieferte MITS auch zwei Versionen einer Programmiersprache, Altair BASIC 4K und 8K. Sie stammten von zwei jungen Männern namens Paul Allen und Bill Gates. Sie gründeten anschließend eine kleine Software-Firma und blieben bis Ende 1978 in Albuquerque. Danach zog man in die Gegend von Seattle, wo Microsoft noch heute sitzt. Der Altair war also Geburtshelfer des größten Software-Unternehmens der Welt.

Die unerkannte PC-Revolution

It is not unusual to buy machines for reasons other than purely to obtain their functions. Consumer electronics can be part of personal style, personal furniture. And part of the attraction of a range of 'brown goods' like hi-fis and photography equipment can lie in the fact that people play around with them as gadgets. – Leslie Haddon

Dass mit dem Altair 8800 und MITS der Grundstein für die PC-Revolution gelegt wurde, wurde allerdings erst Jahre später klar. Seinerzeit gingen die etablierten Computerfirmen davon aus, dass zukünftige Heimnutzer eher über Terminals verfügen würden, die via Telekommunikationsleitungen an Time-Sharing-Grossrechner in Rechenzentren angeschlossen sind. Selbst Produzenten von Mikroprozessoren wie Intel glaubten nicht an einen Markt für eigenständige „Mikrocomputer", sondern vermarkteten Mikroprozessoren als elektronische Komponenten, die in andere Geräte eingebettet werden, um diese mit Programmierfähigkeit (bzw. „Intelligenz") zu versehen. Sie überliessen die Entwicklung der für sie uninteressanten kleinen Computern rund um die Mikroprozessoren den „grassroot users", also Elektronikbastlern und Hobbyisten, sowie den Vertrieb kleinen Firmen, die Elektronikschaltungen und -geräte für diesen Kundenkreis fertigten.

Die interessierten Bastler hatten meist keine konkreten Anwendungen vor Augen (Textverarbeitung und andere Büroanwendungen kristallisierten sich erst Jahre später, mit besseren Druckern, grösseren und flexibleren Speichermedien sowie lokaler Vernetzung, als Anwendungsgebiete heraus, auch primitive Computerspiele waren anfangs eher ein Privatvergnügen), sondern wollten vor allem einfach ihrem Bastelhobby nachgehen und mit der Technik spielen – so wie die Funkamateure, Radiobastler und Modelleisenbahnenthusiasten. Das schnelle Wachstum dieses Bereichs in den USA überraschte alle. Ein Journalist der bekannten britischen Fachzeitschrift Electronics Weekly erinnert sich:

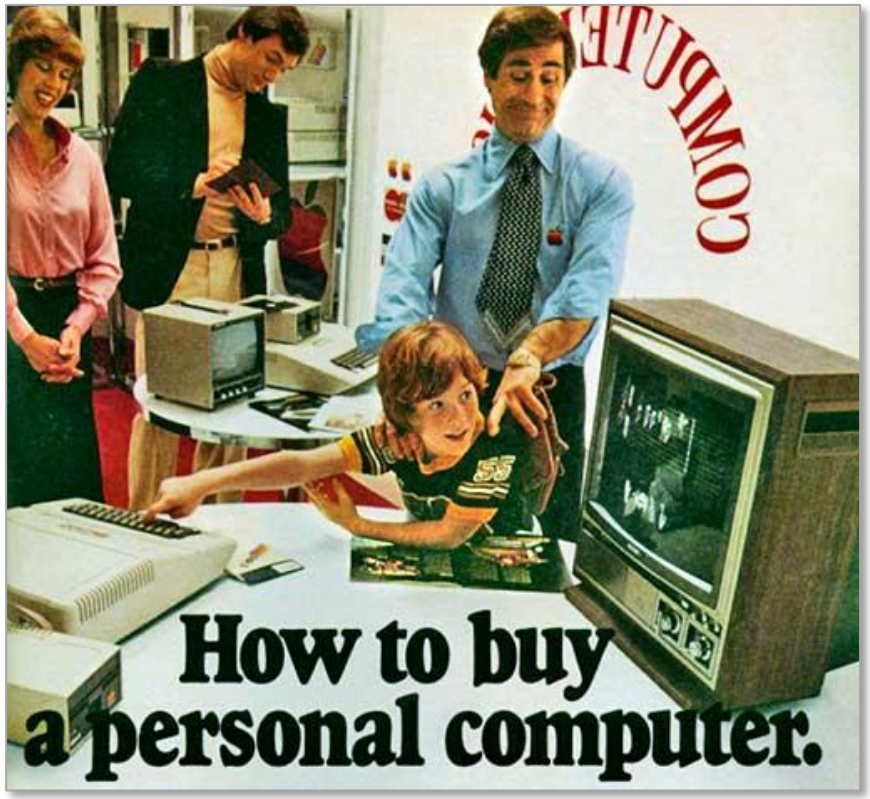
About the time that MITS launched their first machine the editor happened to be in America. He came back with a press cutting from a magazine and he was very amused and entertained. He said, 'These crazy Americans, what will they do next? They seem to think that people are going to buy computers for their own entertainment!' He then said, 'What's really strange is that Americans, being crazy some of them are! Of course, they have too much money.' [Leslie Haddon]

Sinnvolle Anwendung verzweifelt gesucht

1984

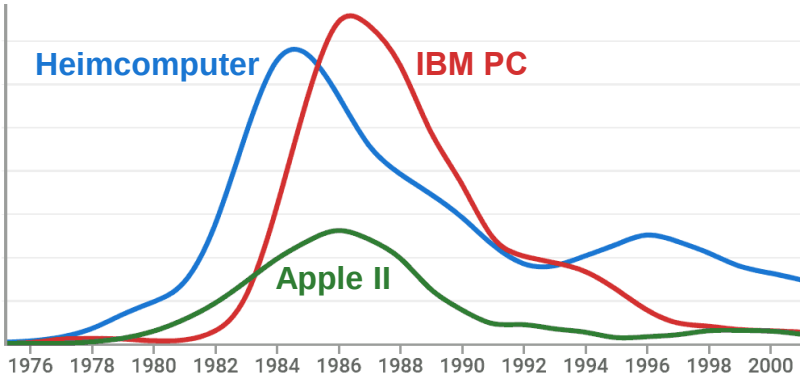
Stiftung Warentest →

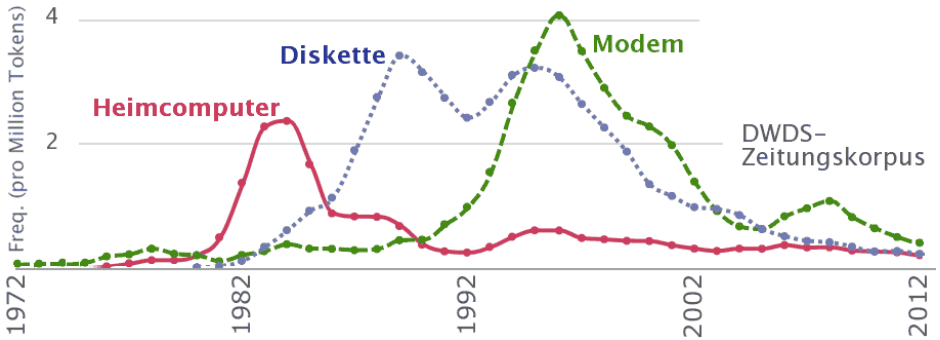
Dass das Verbrauchermagazin damit aus heutiger Sicht grandios falsch lag, ist offensichtlich. Firmen wie Apple oder IBM waren schon seit Ende der 70er-Jahre im Computergeschäft für Privatkunden tätig. Man musste also kein Visionär sein, um dem Gerät gewisse Zukunftsperspektiven vorherzusagen. -- www.welt.de



Heimcomputer → PC

Kochrezepte im Computer zu speichern ist etwa so effizient wie die Fahrt mit dem Raupenschlepper in den Edeka-Laden. -- Der Spiegel 12/1987





Ab 1977 kamen die ersten serienmässigen und gebrauchsfertigen Heimcomputer auf den Markt. Zu diesen gehörten der TRS-80, der Commodore PET 2001 und der Apple II. Betriebssystem und BASIC als Programmiersprache waren in einem ROM gespeichert; Diskettenlaufwerke oder gar Festplatten kamen erst später hinzu, zunächst konnte man Programme und Daten nur mittels Kassettenrecorder abspeichern und laden. Mit dem Markteintritt des IBM PC im Jahr 1981 (Produktvorstellung im Waldorf-Astoria-Hotel in Manhattan) wurde das Feld der persönlichen Computer für Büro-Anwendungen professionalisiert, und Anfang der 1990er-Jahre setzten sich der IBM PC bzw. dazu kompatible Rechner zunehmend auch im Heimbereich durch. Apple stellte mit dem seinerzeit sehr innovativen Macintosh in einem Teilsegment des Marktes eine Alternative dar. Modems, um sich von zuhause aus via Telefonanschluss mit einer Mailbox oder dem Internet zu verbinden, wurden erst in den 1990er-Jahre ein grosses Thema (der Mosaic-Webbrowser für Microsoft Windows erschien Ende 1993).

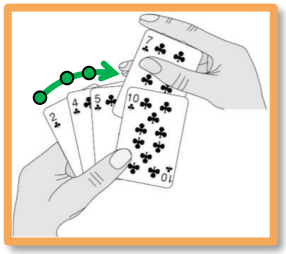
Zeitschriftenanzeige: "*My own IBM computer. Imagine that.*"

1981: Weder Maus noch Fenster, Grafik oder Farbe.

Ende der historischen Notiz

Ein Wettlauf
(Animation)

Achtung, fertig,…



Heapsort

Mergesort

```
// Intervall halbieren: m = n / 2;
// Rekursiv sortieren:
// sort a[1..m]; sort a[m+1..n]
// Sortierte Teil-Arrays „mergen“
```

Insertion sort

```
for (int i = 1; i < n; i++)
{ int t = a[i]; int k;
  for (k = i-1; k >= 0 &&
        a[k] > t; k--)
              a[k+1] = a[k];
  a[k+1] = t;
}
```

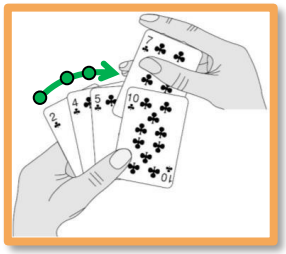
Bekannt aus den Übungen zu Info I

Gleiche Ausgangssituation für alle 3 Algorithmen: 40 Balken der Grösse nach sortieren

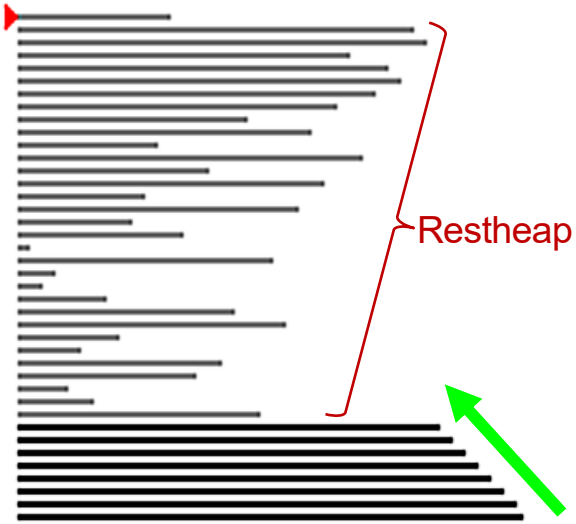
www.sorting-algorithms.com

Ein Wettlauf

➔ **Schnappschuss nach 4 s:**

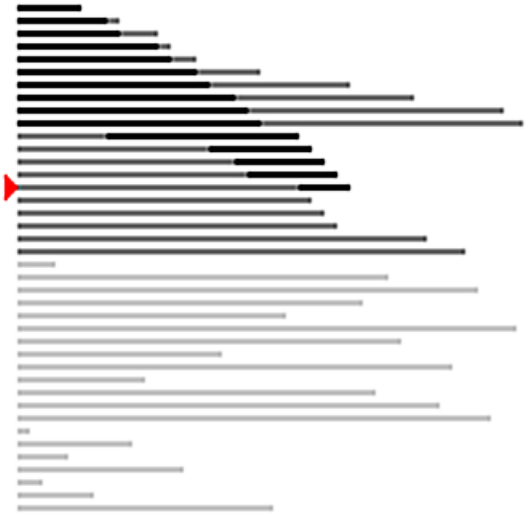


Heapsort



- Rotes Dreieck: aktuell betrachtete Stelle des Array
- Graue Werte: unsortiert
- Schwarze Werte: sortiert

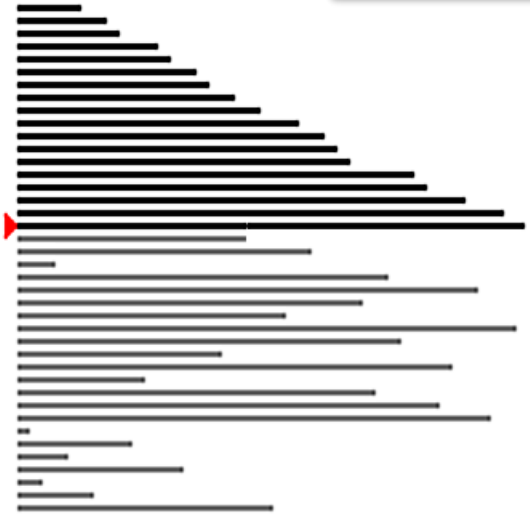
Mergesort



```
// Intervall halbieren: m = n / 2;
// Rekursiv sortieren:
// sort a[1..m]; sort a[m+1..n]
// Sortierte Teil-Arrays „mergen“
```

- Dunkelgrau: aktuelles Intervall bei Mergesort

Insertion sort



```
for (int i = 1; i < n; i++)
{ int t = a[i]; int k;
  for (k = i-1; k >= 0 &&
        a[k] > t; k--)
              a[k+1] = a[k];
  a[k+1] = t;
}
```

Bekannt aus den Übungen zu Info I

Ein Wettlauf

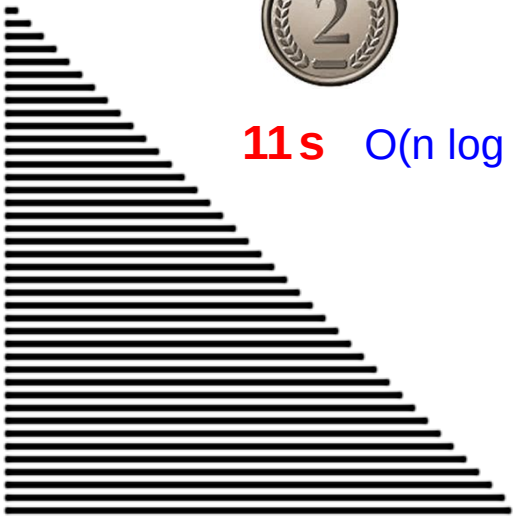
Heapsort

10 s O(n log n)

Da die Speicherzugriffe bei Heapsort recht „wild“ sind und kaum Lokalität aufweisen, kommt es in der Praxis bei grossen Datenmengen allerdings zu Effizienzverlusten durch „cache misses“

- Rotes Dreieck: aktuell betrachtete Stelle des Array
- Graue Werte: unsortiert
- Schwarze Werte: sortiert

Mergesort

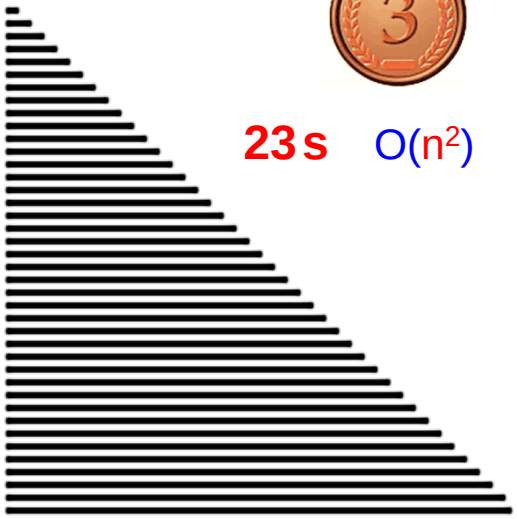


11 s O(n log n)

```
// Intervall halbieren: m = n / 2;
// Rekursiv sortieren:
// sort a[1..m]; sort a[m+1..n]
// Sortierte Teil-Arrays „mergen“
```

- Dunkelgrau: aktuelles Intervall bei Mergesort

Insertion sort



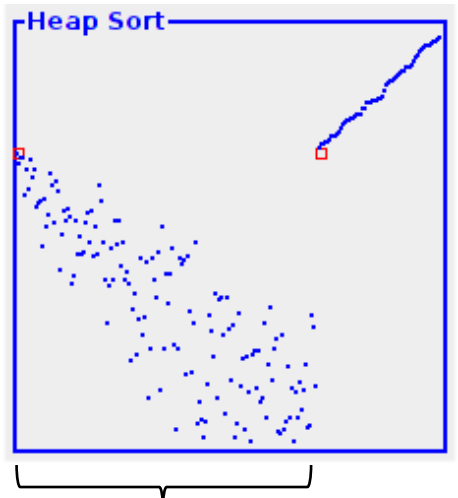
23 s O(n^2)

```
for (int i = 1; i < n; i++)
{ int t = a[i]; int k;
  for (k = i-1; k >= 0 &&
        a[k] > t; k--)
              a[k+1] = a[k];
  a[k+1] = t;
}
```

Bekannt aus den Übungen zu Info I

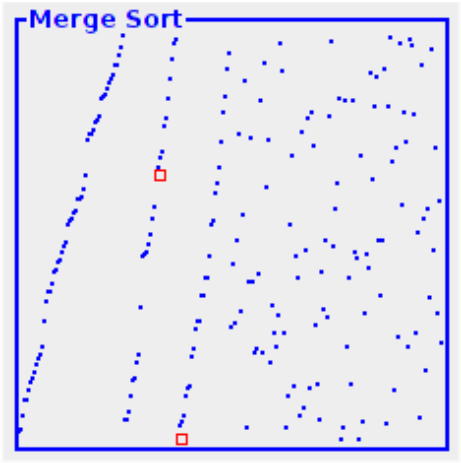
Ein Wettlauf

Heapsort



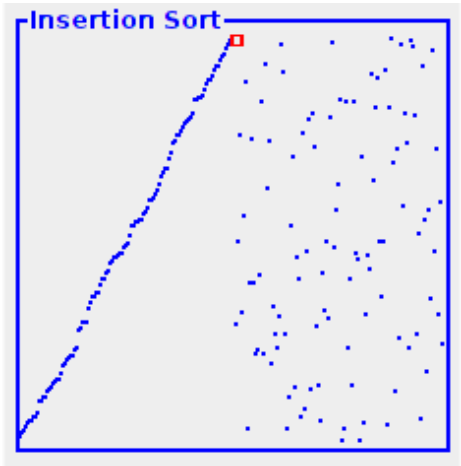
Vorverarbeitet
im Heap

Mergesort



Restmenge noch
unangetastet

Insertion sort



Restmenge noch
unangetastet

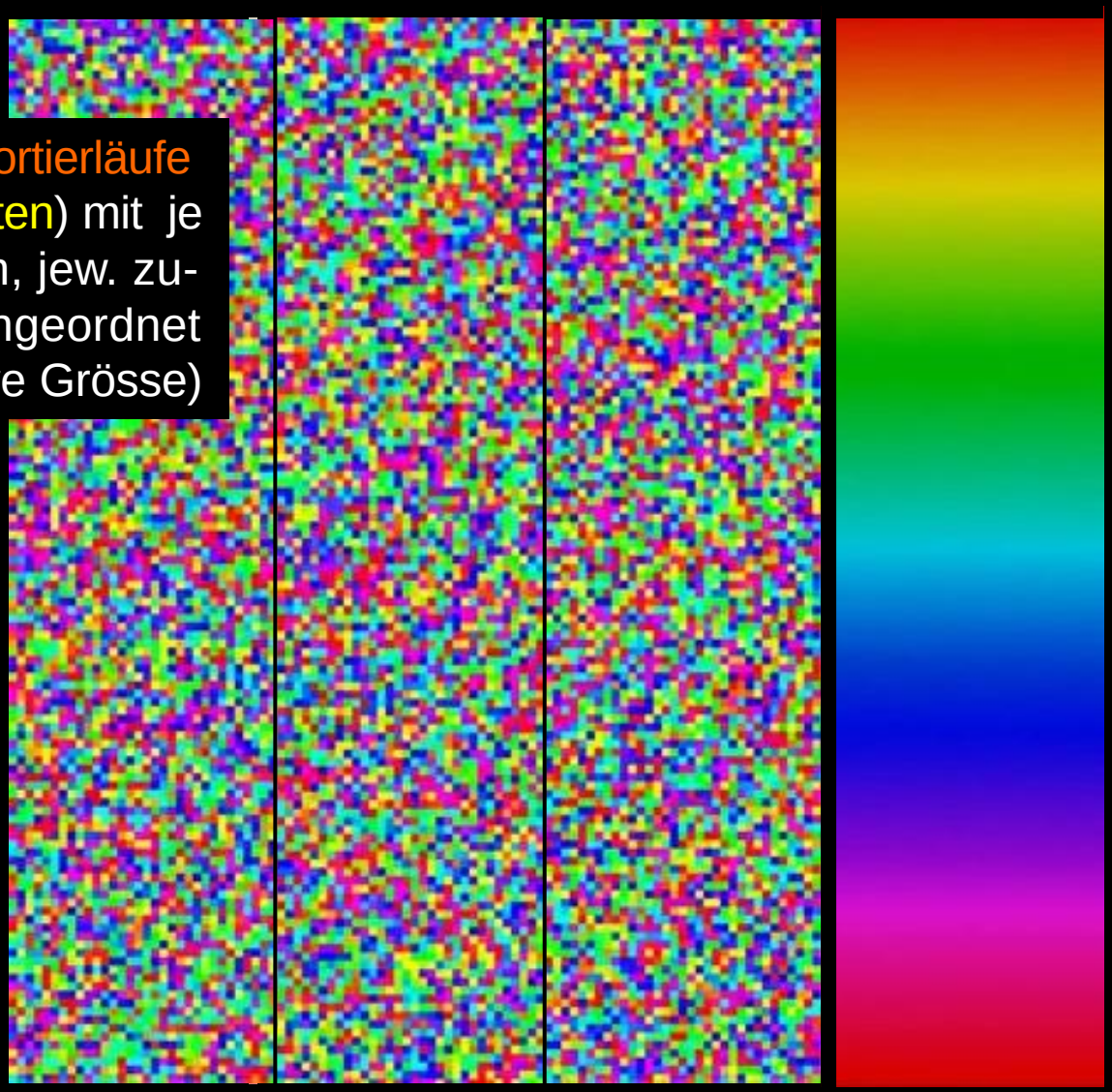
Eine etwas andere Art der Visualisierung durch Schnappschüsse

Bei Heapsort ist die noch zu sortierende Restmenge bereits vorver-
arbeitet und liegt „halbwegs“ (umgekehrt!) sortiert in Heapform vor

Animation: https://i.imgur.com/rWFaMvP.mp4

So sieht das Resultat aus

Quicksort Heapsort Mergesort



35 simultane Sortierläufe (parallele Spalten) mit je 150 Elementen, jew. zufällig anders angeordnet (Farbe = relative Grösse)

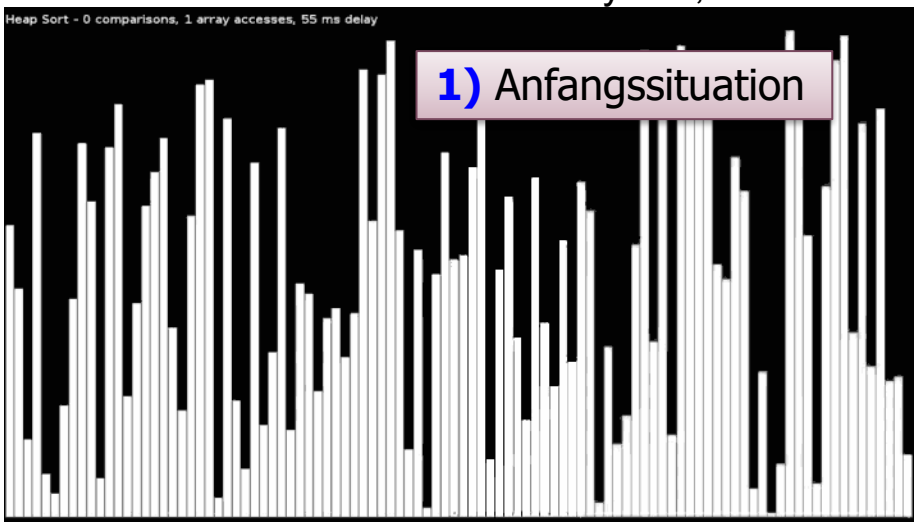
Tipp: Ebenfalls nett anzusehen ist die Sortierscheibe www.youtube.com/watch?v=y9Ecb43qw98

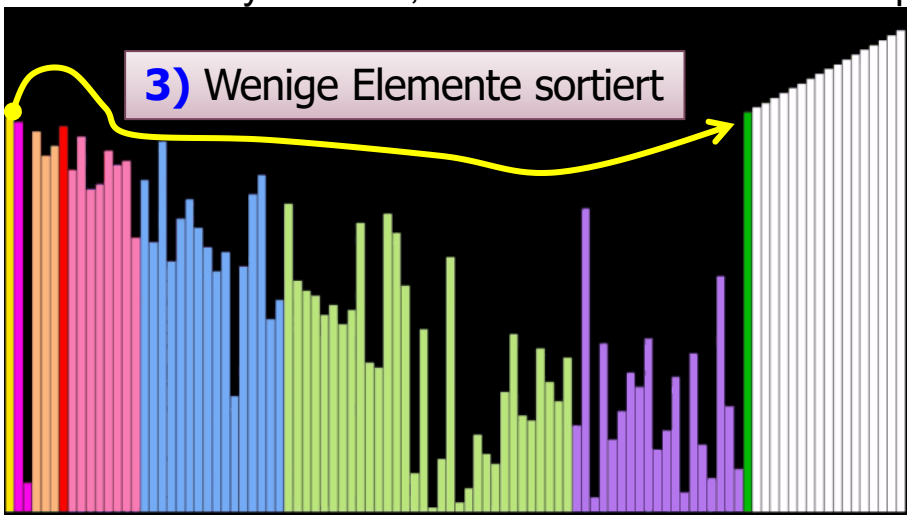
The Sound of Heapsort

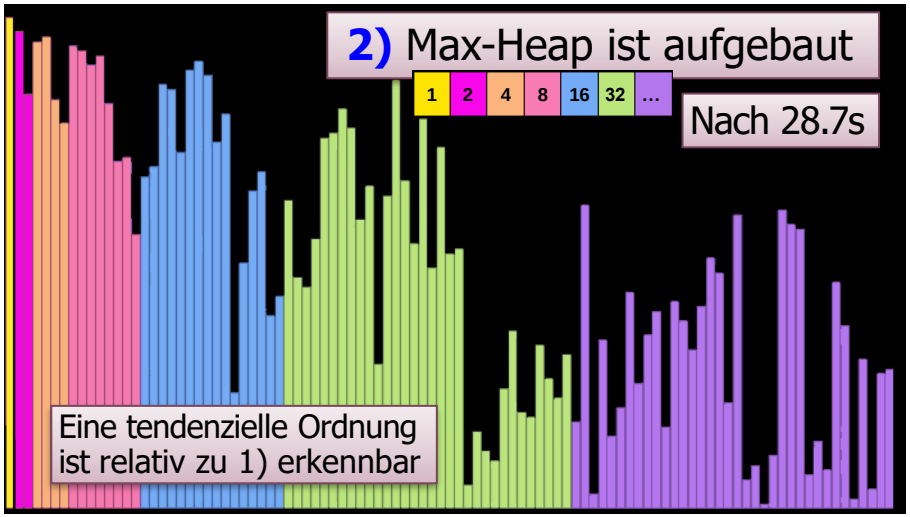
Timo Bingmann, KIT
www.youtube.com/watch?v=_bkow6IykGM

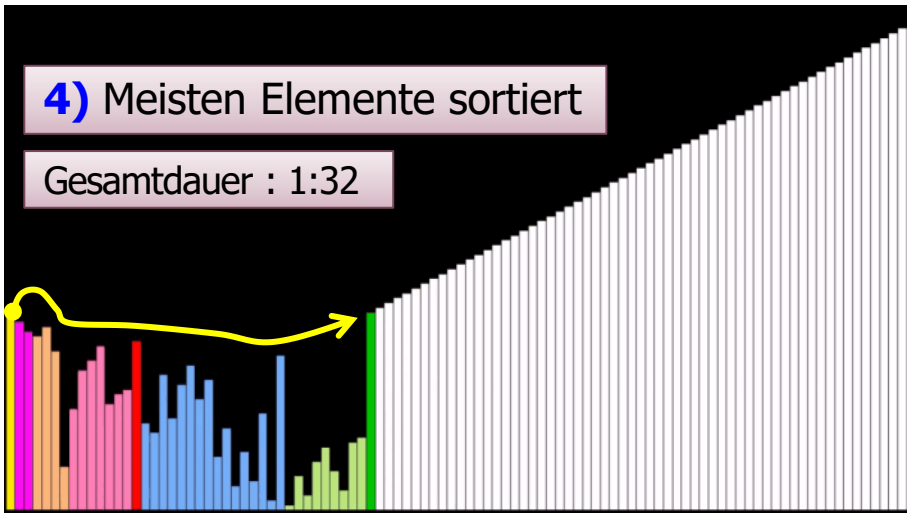
Sortieren von 100 Balken der Länge 1,..,100

Weiss: unsortierter / sortierter Array-Teil; rot: aktuell betrachtete Array-Position; div. Farben: Niveau im Heap.









Zeitkomplexität bekannter Sortierverfahren

Algorithmus	Best case	Average case	Worst case	Stabil?
Insertion sort	O (n)	O (n^2)	O (n^2)	ja
Selection sort	O (n^2)	O (n^2)	O (n^2)	nein
Bubble sort	O (n)	O (n^2)	O (n^2)	ja
Binary tree sort	O (n log (n))	O (n log (n))	O (n^2)	ja
Mergesort	O (n log (n))	O (n log (n))	O (n log (n))	ja
Quicksort	O (n log (n))	O (n log (n))	O (n^2)	nein
Heapsort	O (n log (n))	O (n log (n))	O (n log (n))	nein

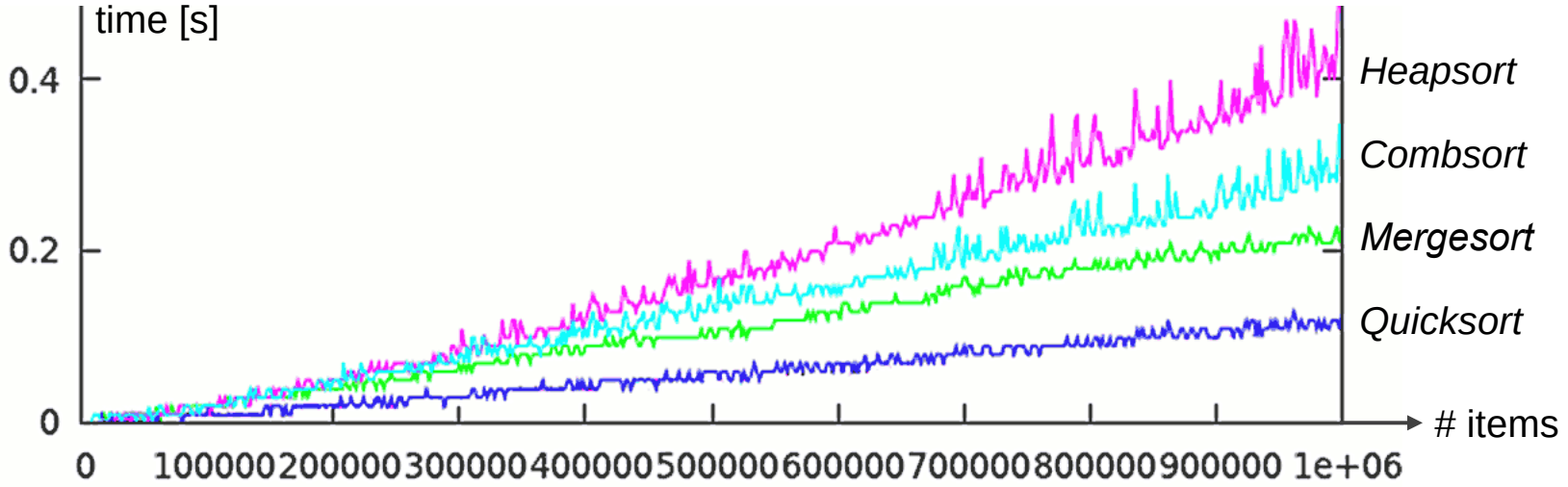
Einige Verfahren wurden schon vor dem Computerzeitalter verwendet (z.B. insertion sort oder selection sort); Quicksort und Heapsort dagegen sind echte Kinder des Computerzeitalters!

Ein stabiles Sortierverfahren bewahrt die Reihenfolge derjenigen Datensätze, deren Sortierschlüssel gleich sind.

Bei bubble sort wird pro Phase die Folge der Elemente von links nach rechts durchlaufen. Dabei wird in jedem Schritt das aktuelle Element mit dem rechten Nachbarn verglichen; falls die beiden Elemente das Sortierkriterium verletzen, werden sie vertauscht. Solange die Folge nicht sortiert ist, wird eine weitere Bubble-Phase initiiert.

Visualisierung bubble sort [Wikipedia]

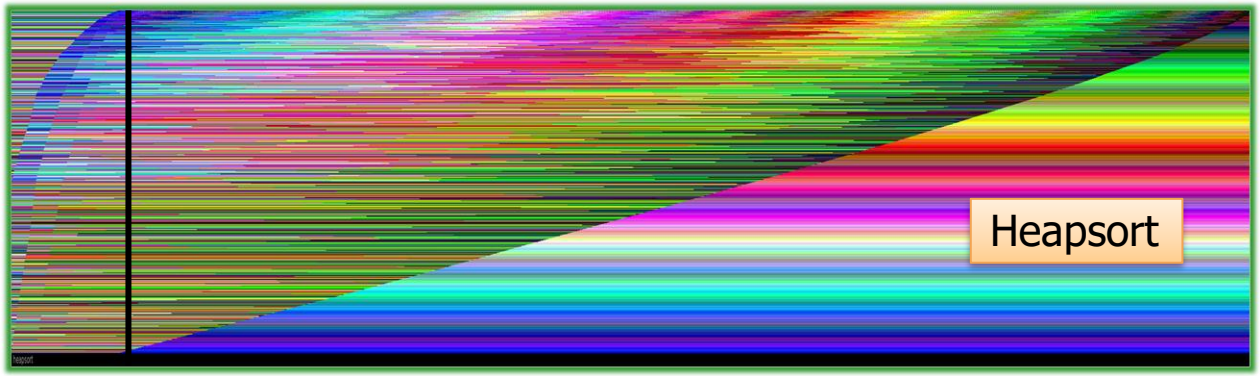
Zeitbedarf von Sortierverfahren in der Praxis

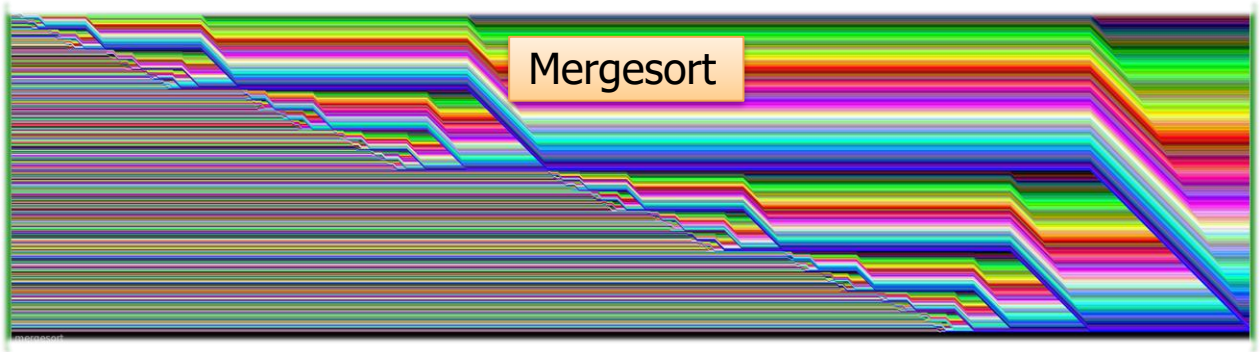


In der Praxis hängt der konkrete Zeitbedarf von der „richtigen“ Implementierung ab, aber auch von den typischen Eigenschaften der Daten (eher eine Zufallspermutation oder oft in Teilen schon vorsortiert?) und sogar von der Computerarchitektur. Heapsort zum Beispiel springt im Speicher wild hin und her, andere Sortierverfahren halten sich eher an das Lokalitätsprinzip – wenn die Daten nicht in den Cache passen, kann sich dies bei Heapsort, wenn keine speziellen Gegenmassnahmen getroffen werden, daher sehr negativ auf die Leistung auswirken.

Obiges Bild als Resultat einer Serie von Experimenten stammt von http://c0de-x.com/i-have-to-sort-that-out/; dort heisst es u.a.: “I didn’t care much about optimizing the code by hand, so don’t expect miracles. I repeated the experiments on increasingly larger data sets. 1,000,000 random integers was the largest dataset I fed them with.”

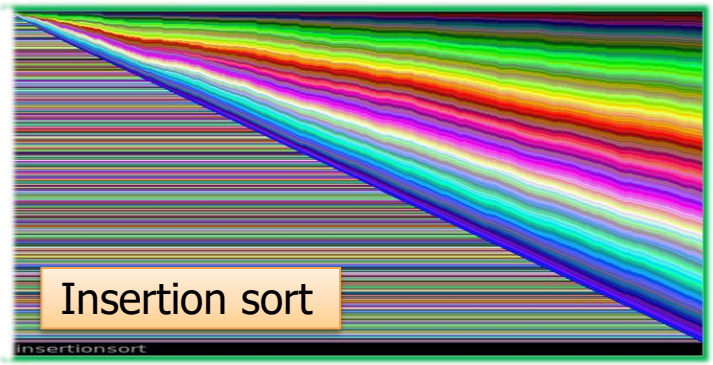
Visualisierung von vier Sortierverfahren

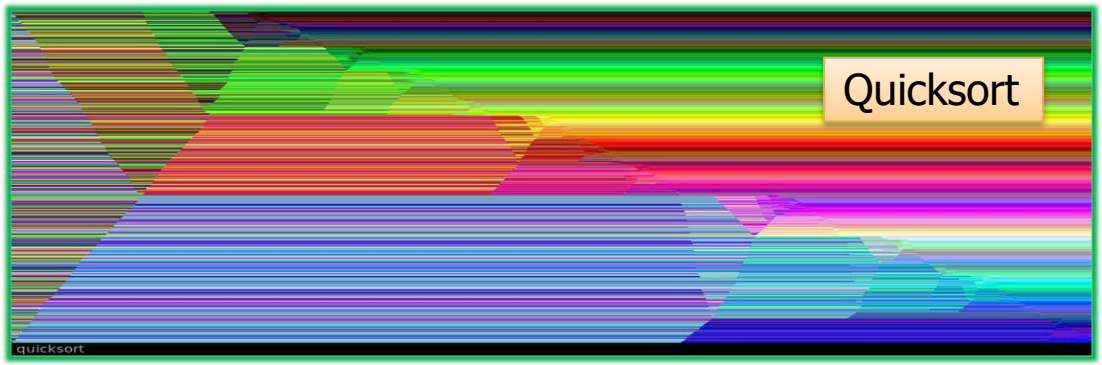




Eine andere Art der Visualisierung, präsentiert von Aldo Cortesi – mehr Informationen dazu bei sortvis.org oder bei https://corte.si/posts/code/visualisingsorting

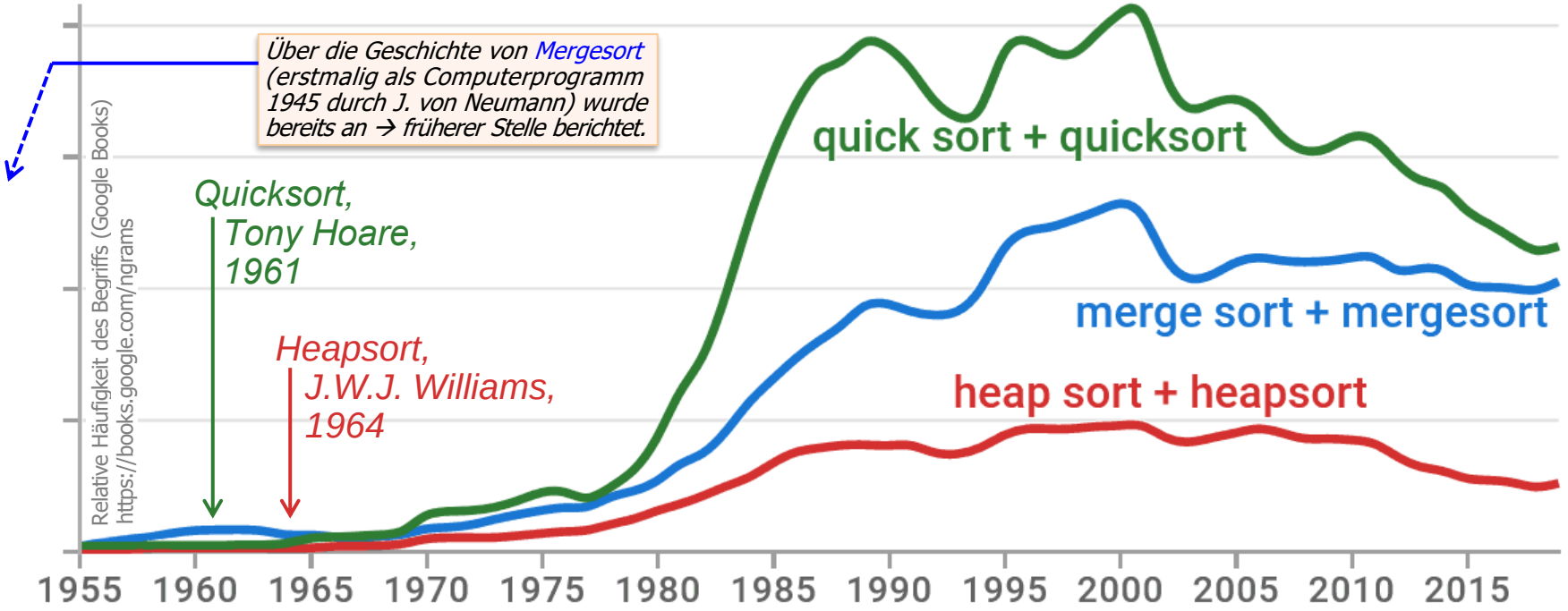
"I dislike animated sorting algorithm visualisations – there's too much of an air of hocus-pocus about them. Something impressive and complicated happens on screen, but more often than not the audience is left mystified. I think their creators must also know that they have precious little explanatory value." -- Aldo Cortesi.





http://sortvis.org/index.html

Popularität guter Sortierverfahren



Quicksort und Heapsort wurden beide in der Zeitschrift „Communications of the ACM“ („CACM“) veröffentlicht, und zwar direkt in der Programmiersprache ALGOL, mit wenigen kurzen Kommentaren dazu.

ALGORITHM 64 — *CACM 7(4), 1961, 321-322*
QUICKSORT
C. A. R. HOARE
Elliott Brothers Ltd., Borehamwood, Hertfordshire, Eng.

```
procedure quicksort (A,M,N); value M,N;
          array A; integer M,N;
comment Quicksort is a very fast and convenient method of
sorting an array in the random-access store of a computer. The
entire contents of the store may be sorted, since no extra space is
required. The average number of comparisons made is 2(M−N) ln
```

ALGORITHM 232 — *CACM 6(7), 1964, 347-348*
HEAPSORT
J. W. J. WILLIAMS (Recd 1 Oct. 1963 and, revised, 15 Feb. 1964)
Elliott Bros. (London) Ltd., Borehamwood, Herts, England

comment The following procedures are related to *TREESORT* [R. W. Floyd, Alg. 113, *Comm. ACM 5* (Aug. 1962), 434, and A. F. Kaupe, Jr., Alg. 143 and 144, *Comm. ACM 5* (Dec. 1962), 604] but avoid the use of pointers and so preserve storage space.

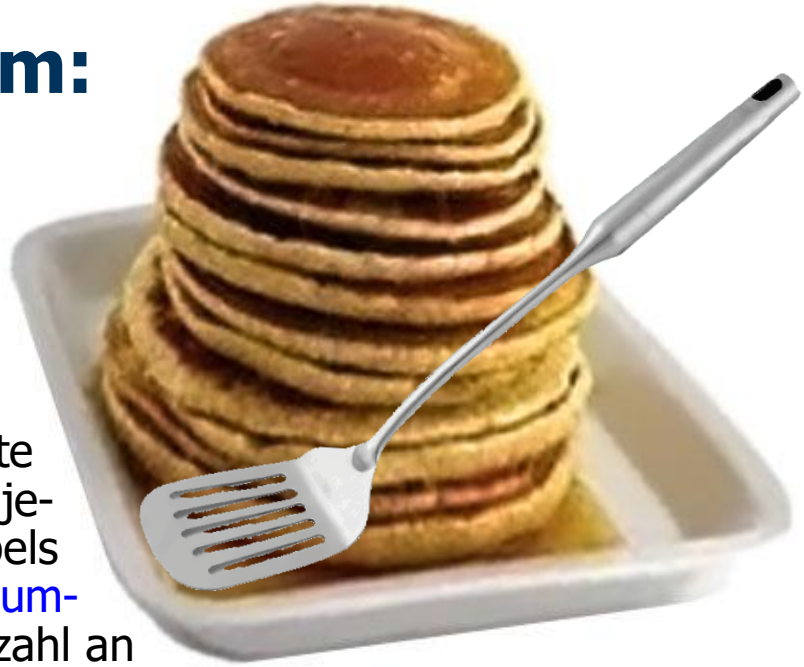
Schneller sortieren mit einem *analogen* Algorithmus?

- Ein wesentlicher Aspekt beim analogen Rechnen ist, dass dabei physikalische Grössen gemessen werden
  - Man hat zwar kein löchriges Werteraster wie beim Digitalen, aber dafür ist i. Allg. das Messen (und Bearbeiten) der Grössen nicht 100% genau

- Algorithmus zum Sortieren von n Werten $a_1,\ldots,a_n$:
  1) Schneide n rohe Spaghetti auf die Längen $a_1,\ldots,a_n$ zu
  2) Nimm sie in die Hand und stelle sie senkrecht auf einen Tisch
  3) Lockere die Hand etwas
  4) Nimm Spaghetti von oben nach unten weg, notiere dabei die Werte

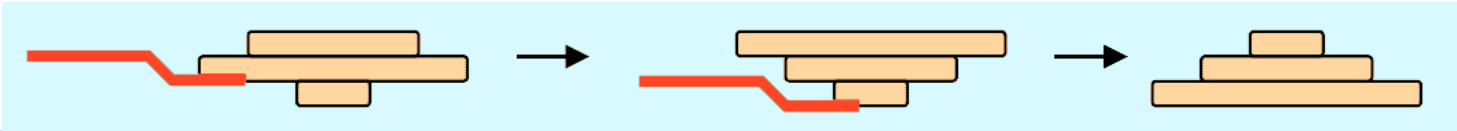
- Laufzeit
  - Geschätzt: 1) 1 min pro Spaghetti; 2) + 3) 10 s; 4) 10 s pro Spaghetti
  - Interessant: Die Gesamtlaufzeit des Verfahrens ist linear in n
  - Es sollte daher alle n log n-Verfahren ab einem gewissen n schlagen, oder?

Und noch ein Sortierproblem: „Pancake sorting“



[Wikipedia:] Beim Pfannkuchen-Sortierproblem geht es anschaulich darum, einen Stapel Pfannkuchen nach der Grösse zu sortieren. Der Stapel soll so sortiert werden, dass am Ende der grösste Pfannkuchen an unterster Stelle ist, der zweitgrösste darüber bis zum kleinsten ganz oben. Dazu darf in jedem Schritt ein von der aktuellen Spitze des Stapels ausgehender Teilstapel ausgewählt und als ganzer umgekehrt werden. Gefragt ist nach der kleinsten Anzahl an Schritten dieser Art, die benötigt werden, um unabhängig von der Ausgangslage den gesamten Stapel zu sortieren. Veröffentlicht wurde das Problem 1975 von Jacob Goodman unter dem Pseudonym Harry Dweighter (reimt mit harried waiter, „gestresster Kellner“):

The chef in our place is sloppy, and when he prepares a stack of pancakes they come out all different sizes. Therefore, when I deliver them to a customer, on the way to the table I rearrange them (so that the smallest winds up on top, and so on, down to the largest at the bottom) by grabbing several from the top and flipping them over, repeating this (varying the number I flip) as many times as necessary. If there are n pancakes, what is the maximum number of flips (as a function of n) that I will ever have to use to rearrange them? [American Mathematical Monthly 82 (1), 1975, pp. 1010]



Bsp. einer Folge von zwei Flips

Pancake sorting mit linearem Aufwand?

Man weiss, dass die Zahl notwendiger Flips für einen Stapel von n Pfannkuchen zwischen 15/14 n (≈ 1.07 n) und 18/11 n (≈ 1.64 n) liegt. (Heisst das, man kann mittels der mächtigen Flip-Operation in linearer Zeit, also O(n), sortieren?)

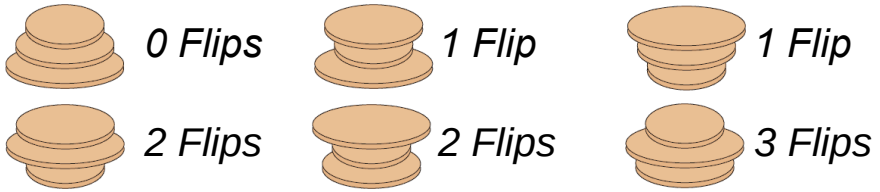
Die britische Zeitung *The Guardian* brachte am 14. Nov. 2013 einen Artikel von Simon Singh zur Geschichte des Pfannkuchen-Sortierproblems; nachfolgend einige Auszüge:

In around 1975, Jacob E. Goodman, a mathematician at the City College of New York, was at home folding towels for his wife. The final pile was somewhat messy, so he decided to restack the folded towels in order of size, smallest folded towel at the top, biggest at the bottom. The problem was that there was no room for a second pile, so he was forced to flip over the top few towels, reassess the situation, then flip over a few more from the top, and so on.

He recalls how a curious problem crossed his mind: "How many flips would I need in the worst case? I thought it was interesting enough to send to the American Mathematical Monthly, but a more 'natural' setting seemed to be pancakes." Thus the so-called pancake sorting problem was born. How many flips are required to turn a disordered stack of pancakes into an ordered stack?

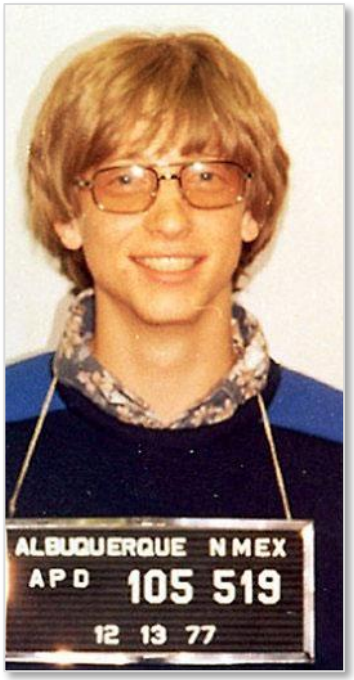
Goodman was still building his reputation as a mathematician and did not want others to think that he was only interested in trivial pancake puzzles, so he adopted a false identity: "It was easy enough to come up with the jocular pseudonym Harry Dweighter ('harried waiter'), but what if the American Mathematical Monthly wanted to contact me in connection with publishing the problem? I told the secretaries at the department that any calls for Harry Dweighter should be diverted to me."

Pancake sorting und Bill Gates

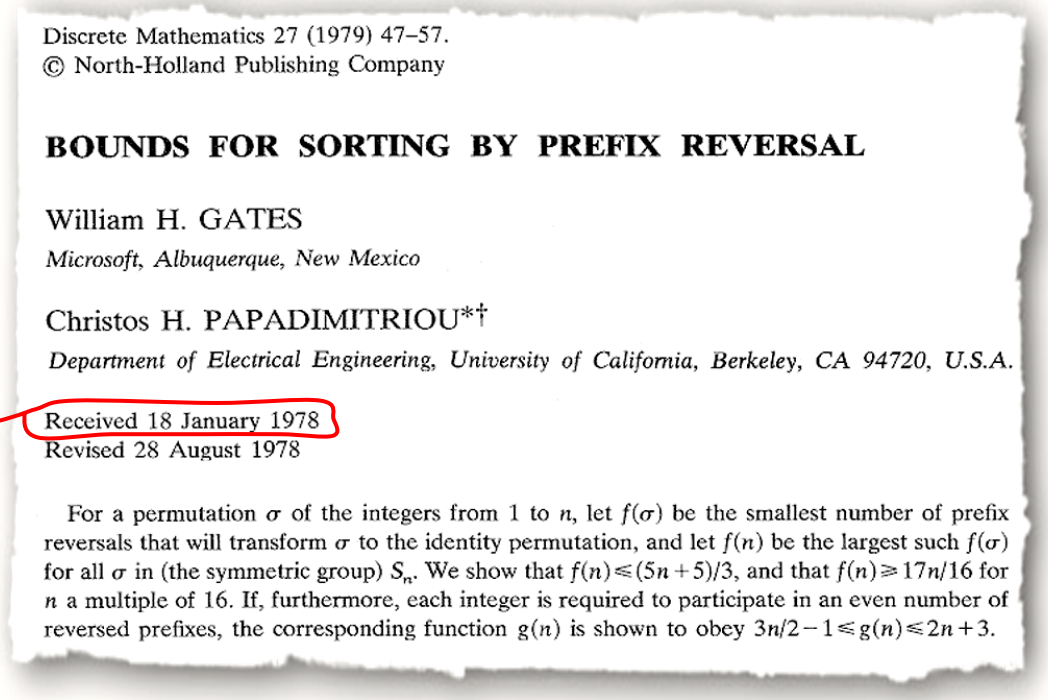


Bei einem Dreierstapel schafft man es stets mit 3 oder weniger Flips, fl(3) = 3. Mit wachsender Stapelgrösse wird das Problem schwieriger, da es mehr Konfigurationen und Flip-Möglichkeiten gibt. Man kennt fl(17) = 19; fl(18) = 20; fl(19) = 22; die genauen Werte ab fl(20) sind aber unbekannt.

Interessanterweise veröffentlichte Bill Gates zusammen mit dem bekannten theoretischen Informatiker Christos Papadimitriou ein wissenschaftliches Paper zu diesem Aspekt [William Gates, Christos Papadimitriou: Bounds for Sorting by Prefix Reversal. Discrete Mathematics, 27 (1) 47-57, 1979]. Die Autoren zeigen, dass (5n+5)/3 eine obere Schranke für fl(n) darstellt.



Das Polizeifoto („mug-shot") von Bill Gates entstand im Dez. 1977; er nahm es seinerzeit in seinem Porsche mit den Verkehrsvorschriften nicht so genau...



Discrete Mathematics 27 (1979) 47–57.
© North-Holland Publishing Company

BOUNDS FOR SORTING BY PREFIX REVERSAL

William H. GATES
Microsoft, Albuquerque, New Mexico

Christos H. PAPADIMITRIOU*†
Department of Electrical Engineering, University of California, Berkeley, CA 94720, U.S.A.

Received 18 January 1978
Revised 28 August 1978

For a permutation σ of the integers from 1 to n, let $f(\sigma)$ be the smallest number of prefix reversals that will transform σ to the identity permutation, and let $f(n)$ be the largest such $f(\sigma)$ for all σ in (the symmetric group) S_n. We show that $f(n) \leq (5n+5)/3$, and that $f(n) \geq 17n/16$ for n a multiple of 16. If, furthermore, each integer is required to participate in an even number of reversed prefixes, the corresponding function $g(n)$ is shown to obey $3n/2 - 1 \leq g(n) \leq 2n+3$.

Christos Papadimitriou über Bill Gates

Papadimitriou recalled: “When I was an assistant professor at Harvard, Bill was a junior. My girlfriend back then said that I had told her: ‘There’s this undergrad at school who is the smartest person I’ve ever met.’ That semester, Gates was fascinated with a math problem called pancake sorting: How can you sort a list of numbers, say 3-4-2-1-5, by flipping prefixes of the list? You can flip the first two numbers to get 4-3-2-1-5, and the first four to finish it off: 1-2-3-4-5. Just two flips. But for a list of n numbers, nobody knew how to do it with fewer than 2n flips. Bill came to me with an idea for doing it with only 1.67n flips. We proved his algorithm correct, and we proved a lower bound—it cannot be done faster than 1.06n flips. […]. It was a silly problem back then, but it became important, because human chromosomes mutate this way. 2 years later, I called to tell him our paper had been accepted to a fine math journal. He sounded eminently disinterested. He had moved to Albuquerque to run a small company writing code for microprocessors, of all things. I remember thinking: ‘Such a brilliant kid. What a waste.’”

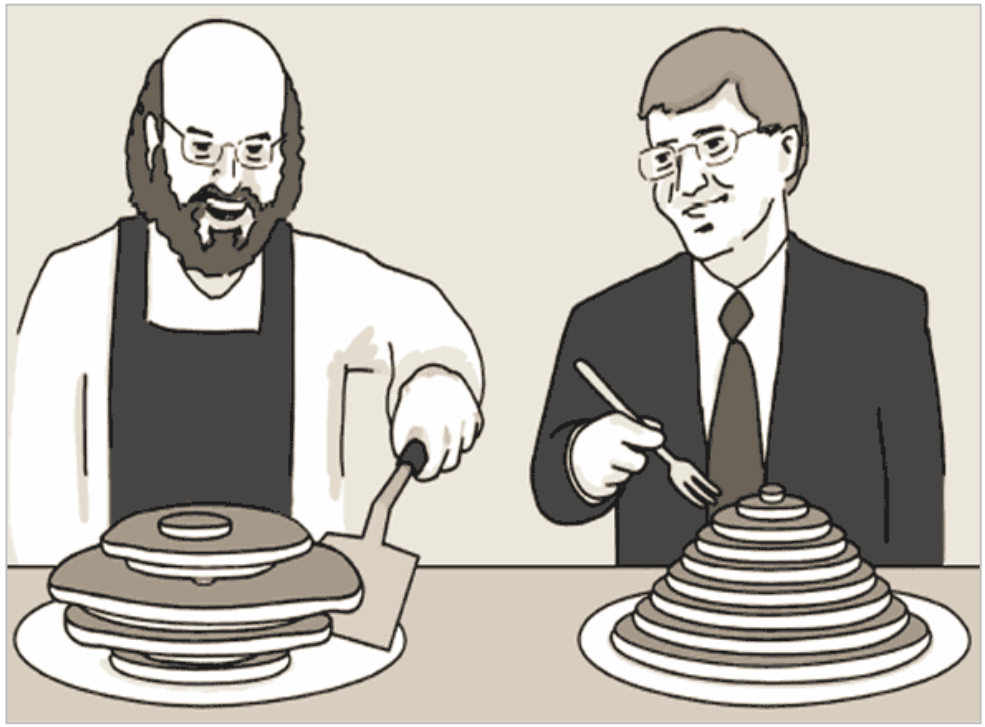


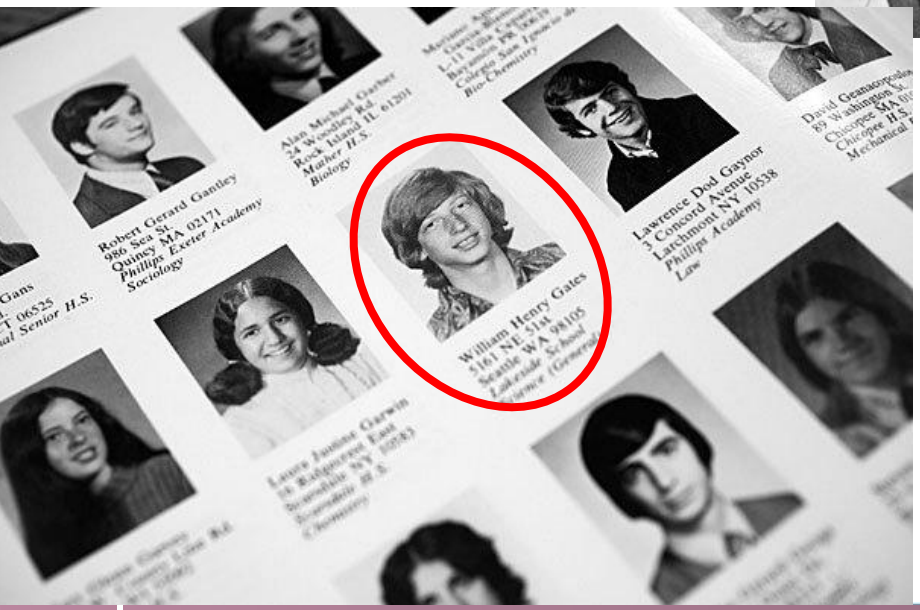
Image source: An Introduction to Bioinformatics Algorithms (Neil Jones, Pavel Pevzner, 2004), p. 130

www.businessinsider.com/a-story-about-bill-gatess-intelligence-2015-11

Young Bill Gates

Bill Gates (13, standing) and Paul Allen (15) connect to a PDP-10 computer at the University of Washington, through a teletype terminal at Lakeside School in 1968. Bill Gates' interest in computers was sparked when he was able to program an early computer to play tic-tac-toe with a human. When he and three others found a way to change the computer's programming so they could get free time on it, they were banned for the summer from school. However, they returned, and were then hired to write first a payroll program on the computer, and then one for scheduling classes. Bill modified the scheduling program so that he was placed in classes with "an inordinate number of interesting girls."

Bild: http://i.imgur.com/EaYMabg.jpg



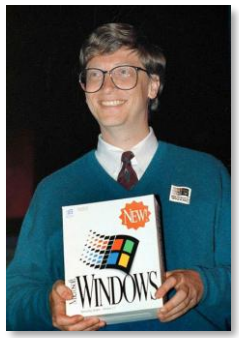
Quelle: http://triogenius.blogspot.com/2014/09/tic-tac-toe-pancakes-and-malaria-bill.html

Links: Bill Gates in seinem "freshman year-book"; er war von 1973 bis 1975 in Harvard. 1975 gründete er mit Paul Allen Microsoft. Dazu mehr bei
→ news.harvard.edu/gazette/story/2013/09/dawn-of-a-revolution/

Bill Gates… und Steve Jobs

https://app.wizer.me/preview/4BJXR

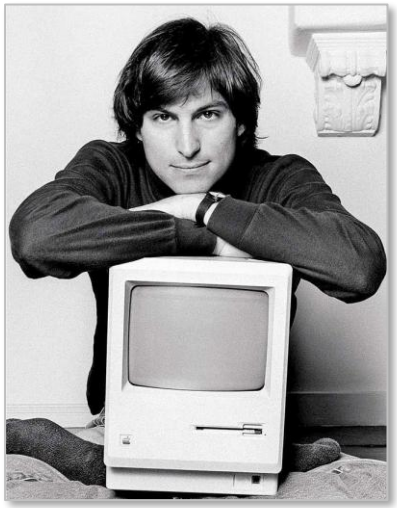
Steve Jobs (li.), der Gründer von Apple, zusammen mit Bill Gates (re.) im Jahr 1991. Beide sind 1955 geboren (Steve Jobs starb 2011), beide haben das Studium abgebrochen und beide „stahlen“ die Idee einer mausgesteuerten und fensterbasierten graphischen Benutzungsoberfläche von der Computer-Workstation Alto der Firma Xerox. Jobs und Gates wurden Rivalen, dominierten mit ihren schnell wachsenden Firmen aber teilweise unterschiedliche Marktsegmente, und sie respektierten einander in späteren Jahren.



Rechts: Ca. 1984.

Links: Bei PC-Betriebssystemen dominiert Microsoft den Markt, Apple ist bei mobilen Geräten erfolgreicher.

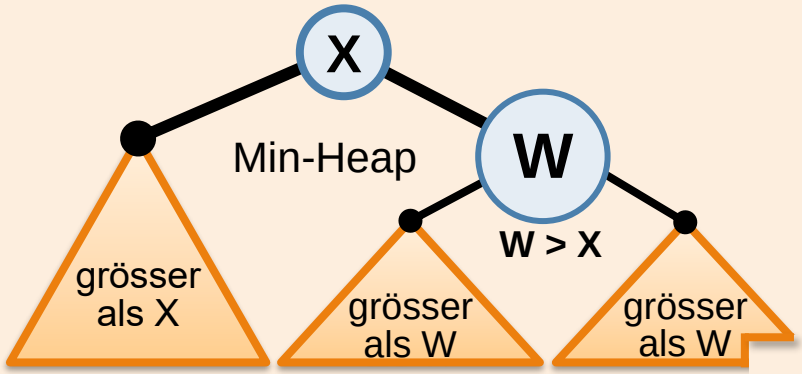
www.epmconsult.it/wordpress/2017/06



http://my-musiclist.com/genius-rivals.html

Resümee des Kapitels

- Heap-Datenstruktur
  - (Fast) vollständiger Binärbaum
  - Werte entlang der Äste sind sortiert
  - *insert*; *get_min* jeweils in O(log n)
  - Operationen besonders effizient bei niveauweiser Speicherung als Array
  - Java-Implementierung der Operationen



- Heapsort
  - O(n log n) Zeitaufwand (selbst im worst case!)
  - „In place" im Array