Chapter 2

Data structures

2.1 Situation (Jim Gray, 97)

Data structures (databases) store ALL data

The New World:

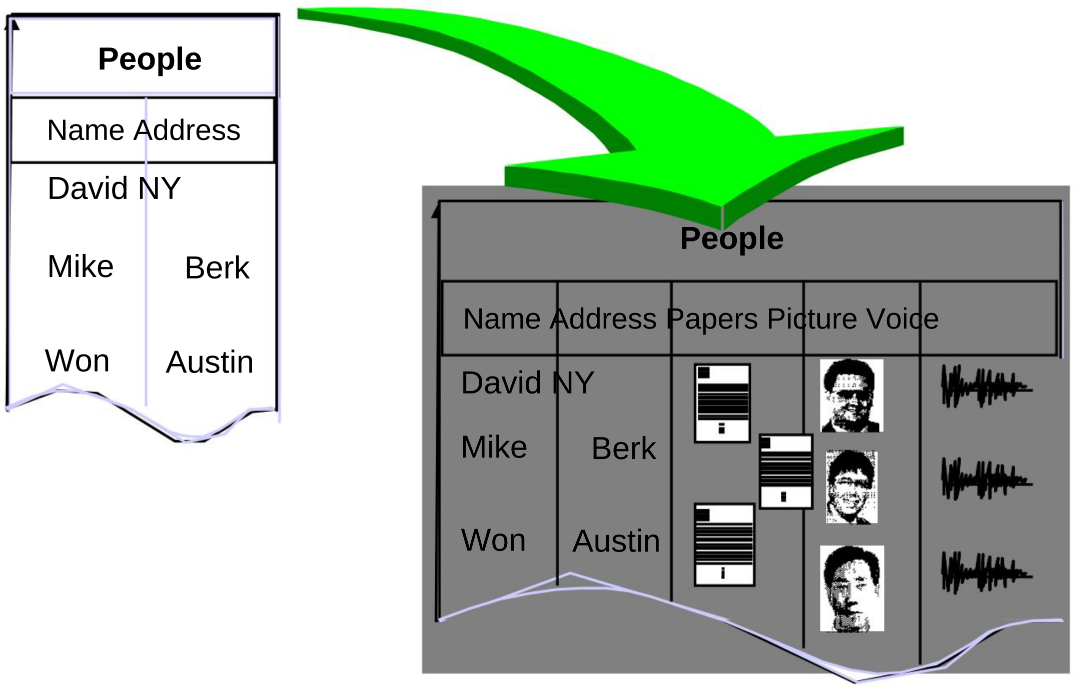
Billions of objects

Large objects (1MB)

The Old World:

Millions of objects

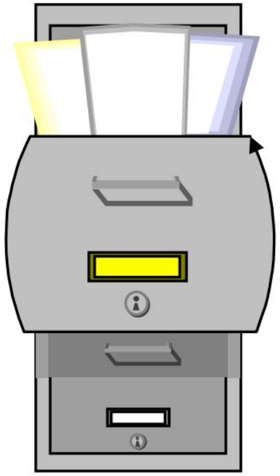
100-Byte Objects



Paperless office
Library of congress online
All information online
entertainment
publishing
business
Information Network,
Knowledge Navigator,
Information at your fingertips

Magnetic disks are cheaper than

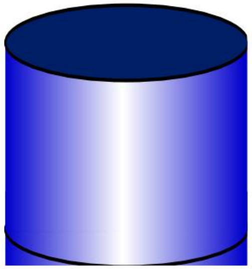
Papier (Jim Gray, 97)



File cabinet:

cabinet (4 drawer) 250$
paper (24,000 sheets) 250$
space (2x3 @ 10$/ft2) 180$
total 700$

3 ¢/sheet



Platte:

disk (4 GB =) 500$
ASCII: 2 m pages
(100x cheaper) 0.025 ¢/sheet

Image: 200 k pages
(10x cheaper) .25 ¢/sheet

Conclusion: Save everything on disks

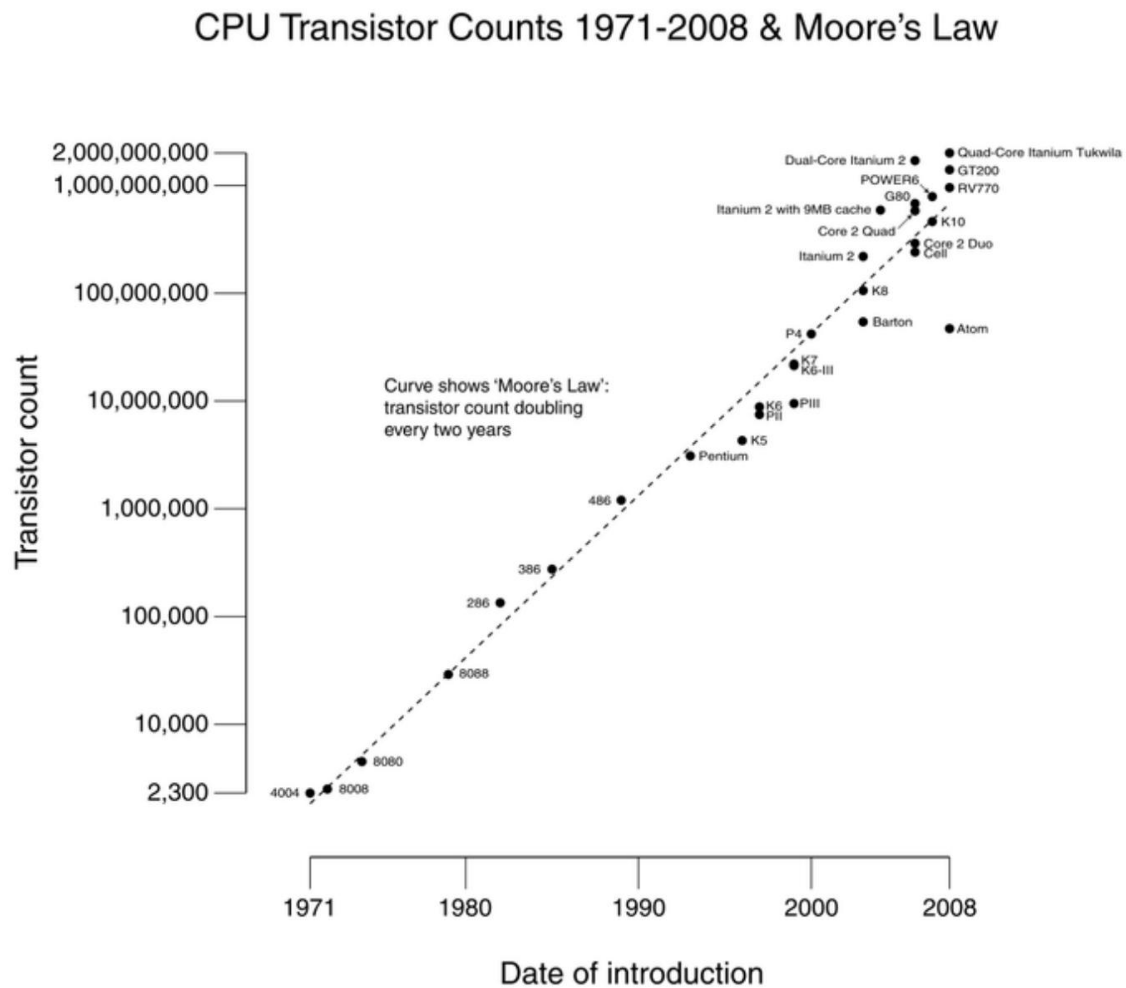
Moore's Law

XXX doubles every 18 (24) months

60% increase per year

- Micro processor speed
- Chip density
- Magnetic disk density
- Communication bandwidth
  - WAN bandwidth
  - approaches LAN

- Disk access
  - speed???



Examples (1)

Magellan Project

Satellite orbiting Venus, radar scanning for surface imaging

Sent 3 terabytes of data

Rendering the data required 13 gigabytes / sec

Was technically not feasible back then

(1994)! http://www2.jpl.nasa.gov/magellan/

Weather

forecast circulation models of the atmosphere and oceans 1000 years simulation, 150 km2 resolution, 0.2 simulated years / machine hour

One run on Intel Touchstone Delta 57 weeks

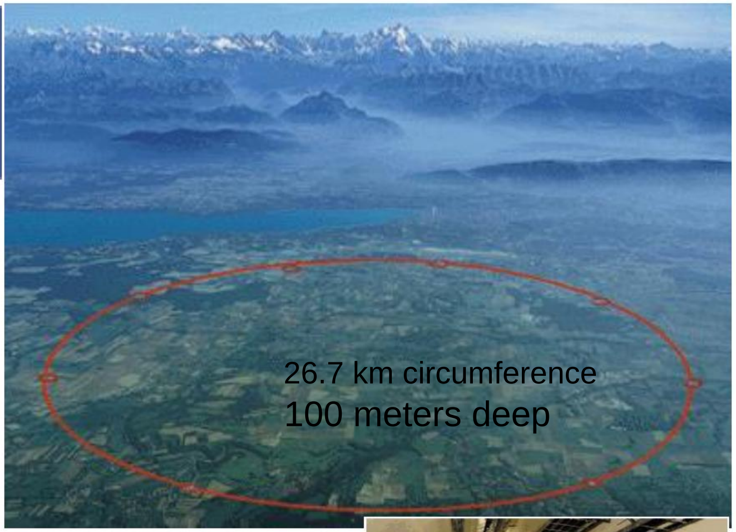
40 MB data / simulation minute = 20 terabytes

Examples (2)

CERN's challenge: data grid

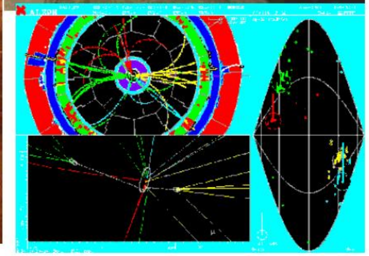
New accelerator LHC with 4 detectors

- Large Hadron Collider, 14 TeV
- Goals: Search for Higgs Boson and Graviton (et al.)
- Start 2009



Goals

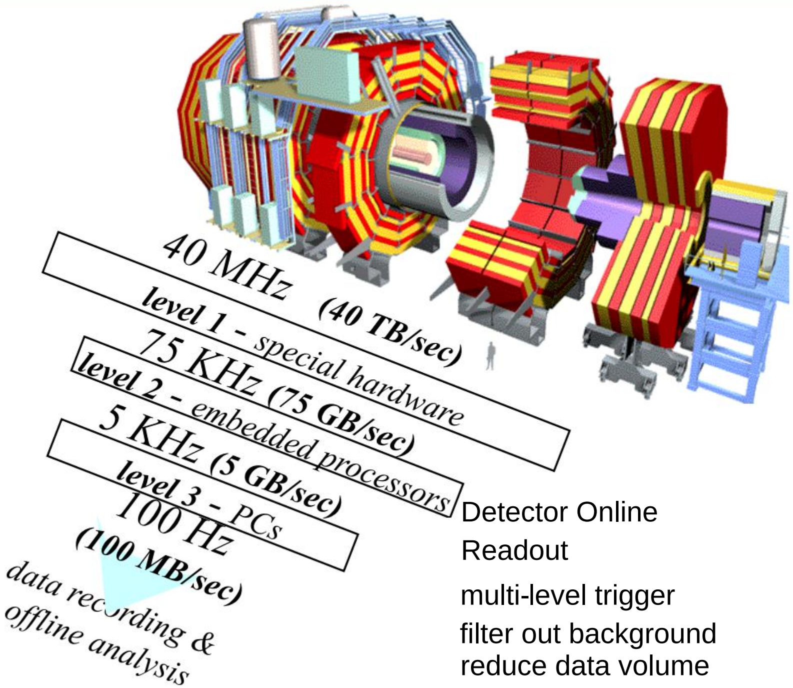
- Worldwide access to the data
  - CERN and Regional Centers (Europe, Asia, America)
  - 2000 users
- Huge data volumes
- Data semantics
- Performance and throughput



Gigantic amounts of data

Characteristic sizes

- 1-6 (maybe 100?) petabytes/year
  - Period 15 to 20 years
- CERN Tier 0 / Worldwide (2009)
  - 24000 / 61500 CPUs
  - 5.6/35 PB disk space
  - 22 / 40 PB tapes
  - 340 gigabytes of IO bandwidth



Detector Online Readout

multi-level trigger

filter out background
reduce data volume

Size: What is a petabyte?

universität wien

1 Petabyte = 2 hoch 50, ie
(1,125,899,906,842,624) bytes ÿ 1015 bytes

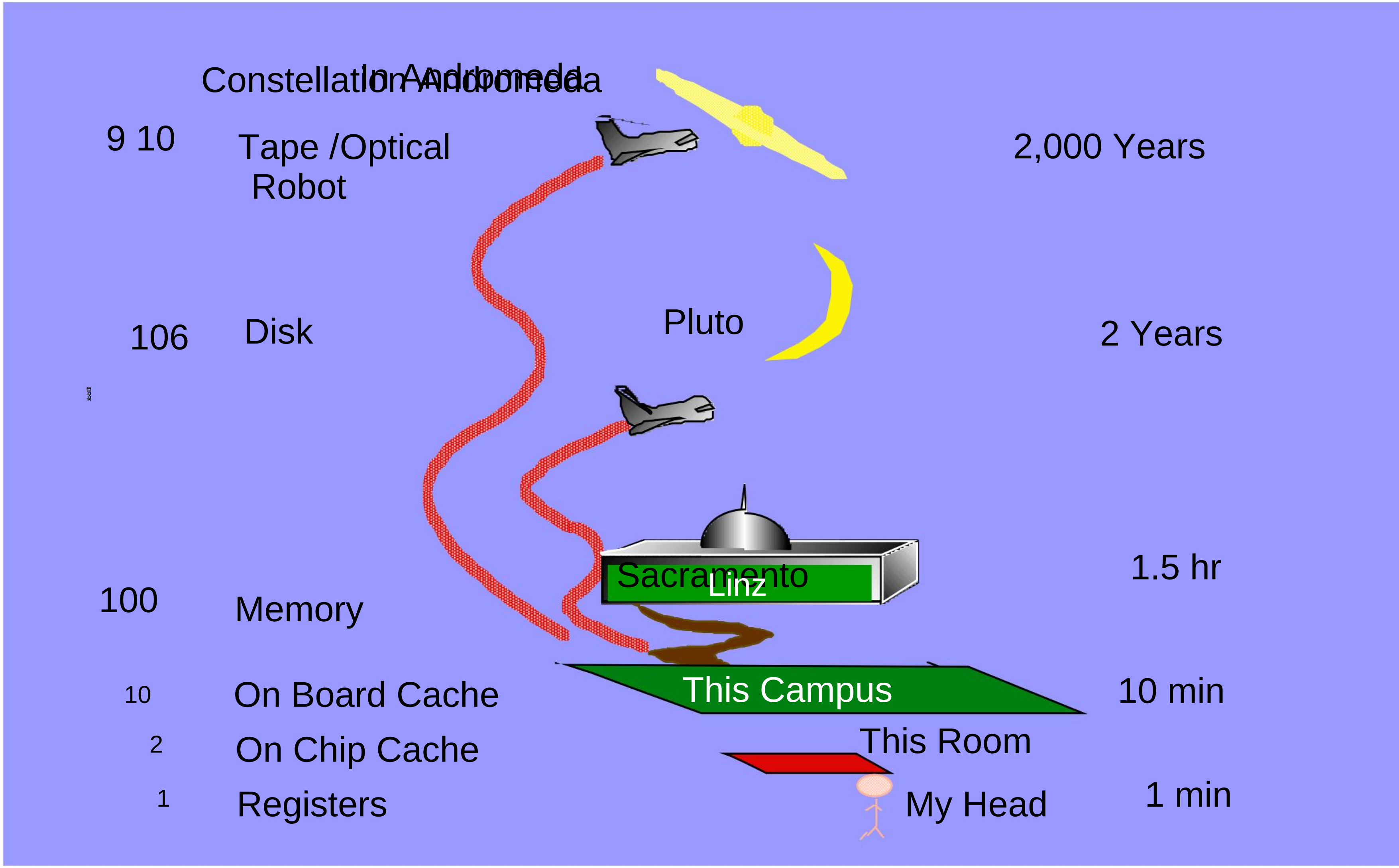
1,000,000,000,000 business letters
100,000,000,000 book pages
50,000,000,000 FAX images
10,000,000,000 TV pictures (mpeg)
4,000,000 LandSat images

150,000 miles of bookshelf
15,000 miles of bookshelf
7,000 miles of bookshelf
10,000 days of video

Library of Congress (in ASCII)
enthält 0.025 Petabyte

Current and future projects generate far more data. Exa, Zeta, Yotta bytes are waiting for us!!!

Speed: memory access times How far away is the data? (Jim Gray, 97)



2.2 Motivation

Example: 100 phone numbers to manage

A "pile" of pieces of paper with names and numbers

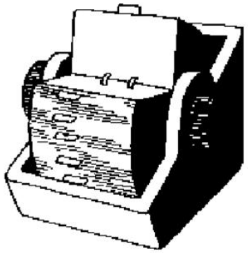
- Finding a phone number through sequential search requires
- an average of 50 "hits"

Sheets sorted by name

- Search by binary splitting ("in the middle, key comparison and then continue searching to the left or right of it")
- About ld 100 ÿ 7 hits

Rolodex

- Pieces of paper sorted, in folders and with name
- index target with one (1!) Access desired number



Goals

Manage information “efficiently”!

What does “efficient” mean?

Quantitative goals

Access time

quick insertion, modification, deletion, ... (i.e. “editing” in the broadest sense) of the data

Storage space

compact storage of information

Qualitative targets

Support for specific types of access to properties or Characteristics of the data

Data structures

Fulfillment of these goals led to the development of

Data structures

Data structures are used to manage large amounts of similar ones Objects

Different data structures are used to manage different Objects that are characterized by different properties, it follows:

Different data structures for different problems!

Example: telephone book management

universität wien

Search in a phone book with 2,000,000 entries

Assumption: access to a data set in 0.01 ms

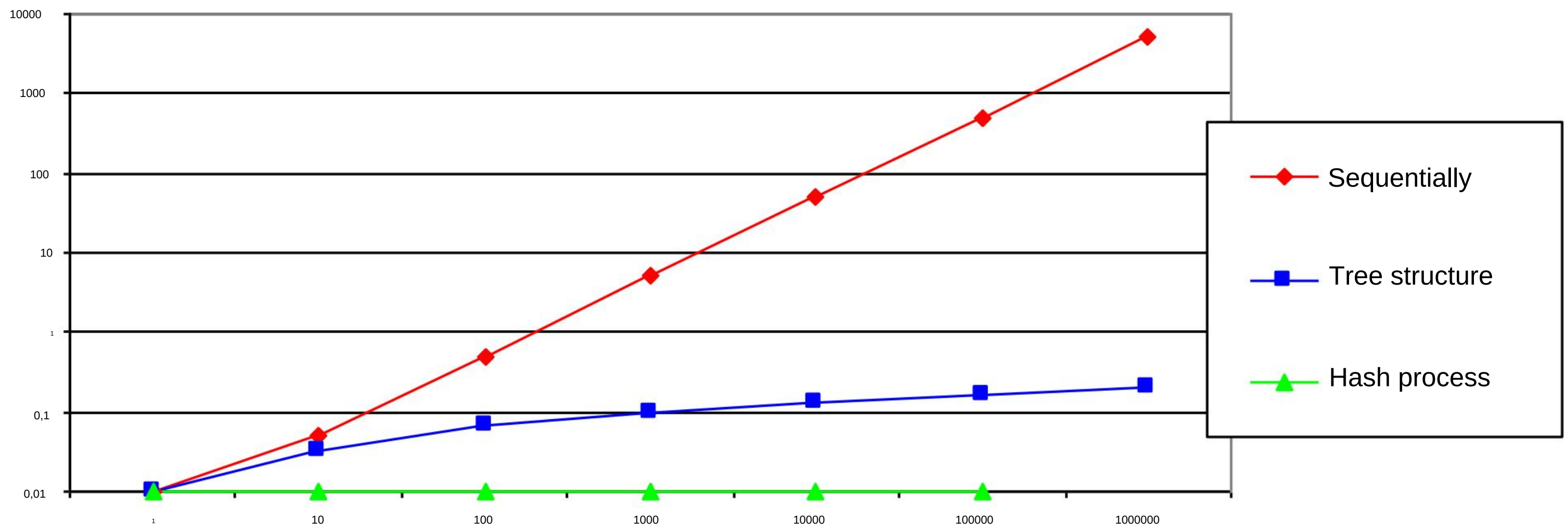
Approaches (time to access)

Sequential search (on average 1000000 * 0,01 ms = 10 s)

Tree structure (approximately ld 2000000 * 0,01 = 0,21 ms)

Hash management (1 access = 0.01 ms)

Access time in relation to file size



2.3 Overview

All known forms of data organization are based on some

a few simple techniques

Sequential techniques

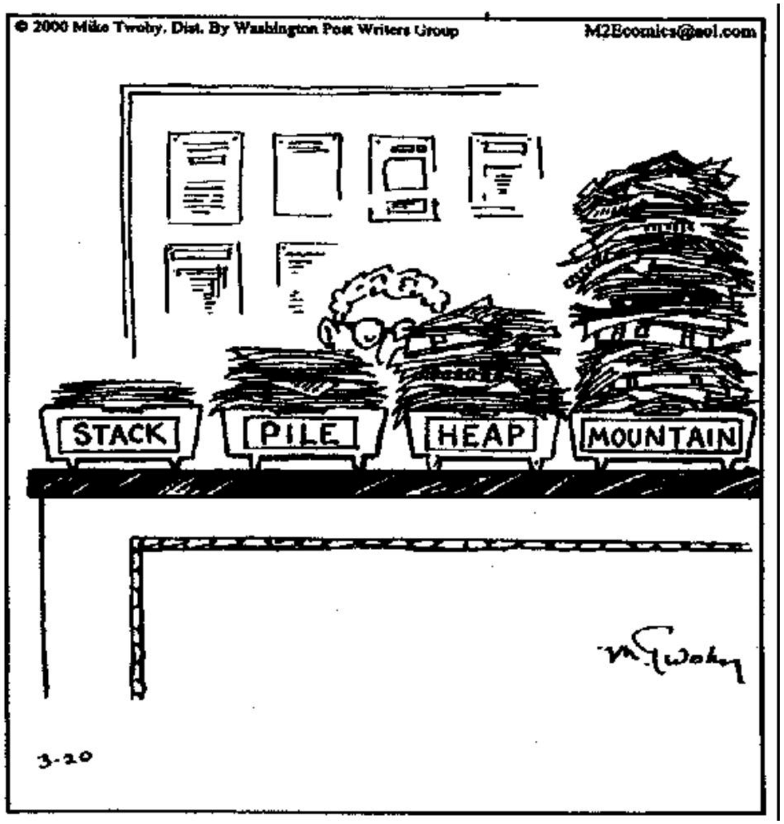
- Listen
- Stack, Queue

Hash process

- Dictionary, Hash Tabelle,
  Collision procedures

Tree structures

- Binary tree, B+ tree,
  Priority Queue



Vector

universität wien

A vector (field) manages a fixed number of elements of a uniform type.

Accessing an element via an integer index (the position in Vector)

Access effort for all elements is constant

x0	x1	x2	…	xn-1	xn

Applications

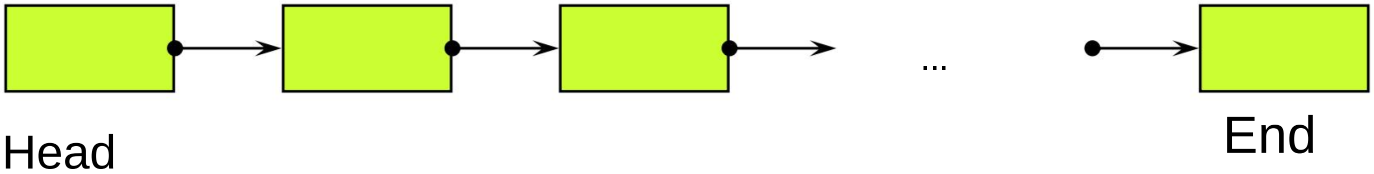
Management of fixed sequences, strings, math. concepts

List

A list is used to manage any number of elements of a uniform type.

The elements are arranged in a sequence that can (usually) be derived from the order of entries (disordered).

The effort involved in accessing an individual element depends on the Position in the sequence.



Applications

sequential data sets, large data sets, external storage

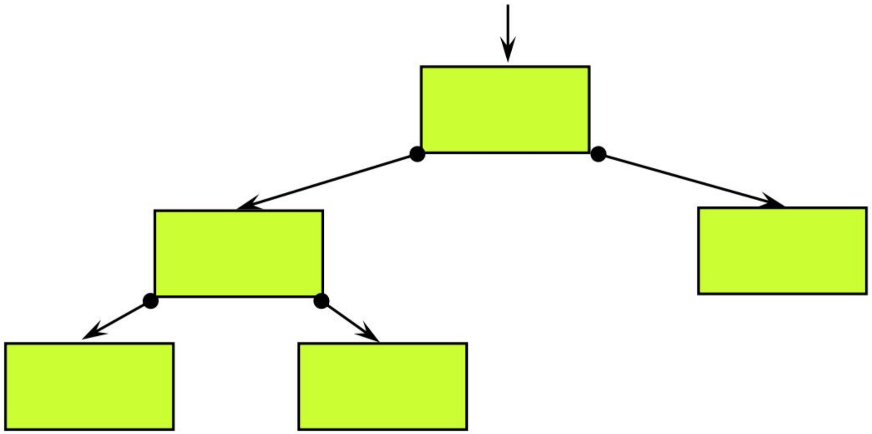
Baum

The tree represents a generalization of the list to a 2-dimensional data structure.

consists of nodes and edges

Exponential relationship between depth of tree and number the knot

Applications: general key management, main and External storage management



Comparison criteria

- dynamics
  - Data management
    - Insert, delete
  - Data set
    - arbitrary or fixed number of elements
- Expense
  - Duration of operations
  - Disk space consumption
- Model
  - Scope of operations

Informal comparison

Data structure	strengths	weaknesses
Vector	dynamic management direct element access constant effort of the The operation low memory	often static (only limited amount of data) limited The operation
List	dynamic management any amount of data clear model	linear effort of the The operation simple Modell
Baum	mostly dynamic Administration any amount of data logarithmic effort of The operation	Balancing algorithms relatively higher Disk space consumption complex model sometimes just insert Operation supported

What do we take with us?

Data organization

Efficiency

Quantity - quality

Data structures

The type

Comparison criteria