

CS 265

*Stratos Idreos*

BIG DATA SYSTEMS

NoSQL | Neural Networks | Image AI | LLMs | Data Science

# the **grammar** of data systems design

# Trillions of possible data structures

Data Calculator @SIGMOD 2018

# Trillions of possible data structures

Data Calculator @SIGMOD 2018

## New NoSQL systems: 1000x faster

Cosine @PVLDB 2022 and Limousine @SIGMOD 2024

# Trillions of possible data structures

Data Calculator @SIGMOD 2018

## New NoSQL systems: 1000x faster

Cosine @PVLDB 2022 and Limousine @SIGMOD 2024

## Synthesized statistics, 10x faster ML

Data Canopy @SIGMOD 2017

# Trillions of possible data structures

Data Calculator @SIGMOD 2018

## New NoSQL systems: 1000x faster

Cosine @PVLDB 2022 and Limousine @SIGMOD 2024

## Synthesized statistics, 10x faster ML

Data Canopy @SIGMOD 2017

## 10x faster Neural Networks

MotherNets @MLSys 2020, and M2 @MLSys 2023

# Trillions of possible data structures

Data Calculator @SIGMOD 2018

## New NoSQL systems: 1000x faster

Cosine @PVLDB 2022 and Limousine @SIGMOD 2024

## Synthesized statistics, 10x faster ML

Data Canopy @SIGMOD 2017

## 10x faster Neural Networks

MotherNets @MLSys 2020, and M2 @MLSys 2023

## 10x faster Image AI

Image Calculator, SIGMOD 2024

## **First two classes:**

Storage is the root of performance for big data systems

We increasingly need new big data systems

Designing systems is super complex (several years and moving targets)

We need to be able to rapidly design performant systems  
....or even fully or partially automate design as much as possible

**what should you be doing?**

**READING**

**preparation**

**Get familiar with the very basics of traditional database architectures:**

Architecture of a Database System. By J. Hellerstein, M. Stonebraker and J. Hamilton. Foundations and Trends in Databases, 2007

**Get familiar with very basics of modern database architectures:**

The Design and Implementation of Modern Column-store Database Systems. By D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden. Foundations and Trends in Databases, 2013

**Get familiar with the very basics of modern large scale systems:**

Massively Parallel Databases and MapReduce Systems. By Shivnath Babu and Herodotos Herodotou. Foundations and Trends in Databases, 2013

**first readings**



## **The Periodic Table of Data Structures.**

Stratos Idreos, Kostas Zoumpatianos, Manos Athanassoulis, Niv Dayan, Brian Hentschel, Michael S. Kester, Demi Guo, Lukas Maas, Wilson Qin, Abdul Wasay, Yiyou Sun.  
IEEE Data Engineering Bull. Sep, 2018

## **Design Continuums and the Path Toward Self-Designing Key-Value Stores that Know and Learn.**

Stratos Idreos, Niv Dayan, Wilson Qin, Mali Akmanalp, Sophie Hilgard, Andrew Ross, James Lennon, Varun Jain, Harshita Gupta, David Li, and Zichen Zhu. Proceedings of CIDR Conference on Innovative Data Systems Research, 2019.

# Preparing for presentations and reviews

Judge lectures  
(content and slides)

## **review and slides should answer:**

- what is the problem
- why is it important
- why is it hard
- why existing solutions do not work
- what is the core intuition for the solution
- solution step by step
- does the paper prove its claims
- exact setup of analysis/experiments
- are there any gaps in the logic/proof
- possible next steps

\* follow a few citations to gain more background

# how to prepare slides

no bullets   2 colors   big text   images   animation for examples

# how to prepare slides

no bullets   2 colors   big text   images   animation for examples

## unified flow

one message per slide   connection from slide to slide   stories

## Today:

Self-designing systems in more detail

Goal: concept, (3) steps needed

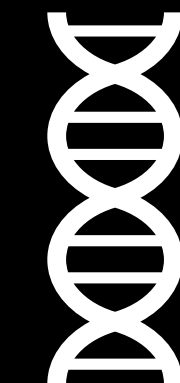
Understanding the first step: design primitives (data structures)

More details on key-values stores

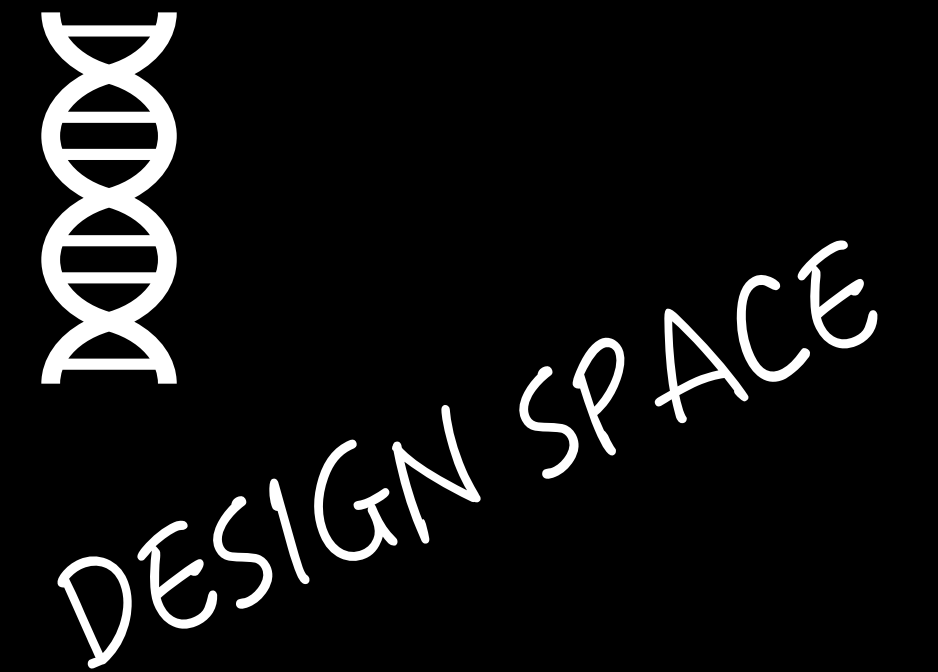
# **4 very high-level ways to present the same thing**

How can we design complex systems automatically?

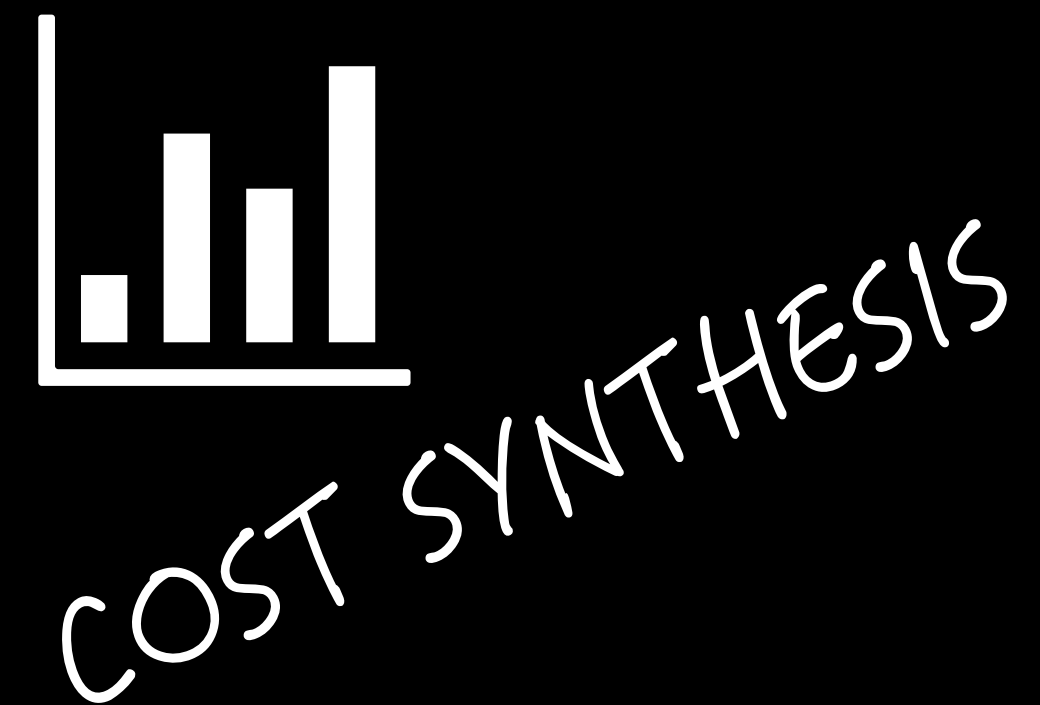
# How many and which designs are possible?

  
DESIGN SPACE

How many and which  
designs are possible?



Can we compute  
performance w/o coding?

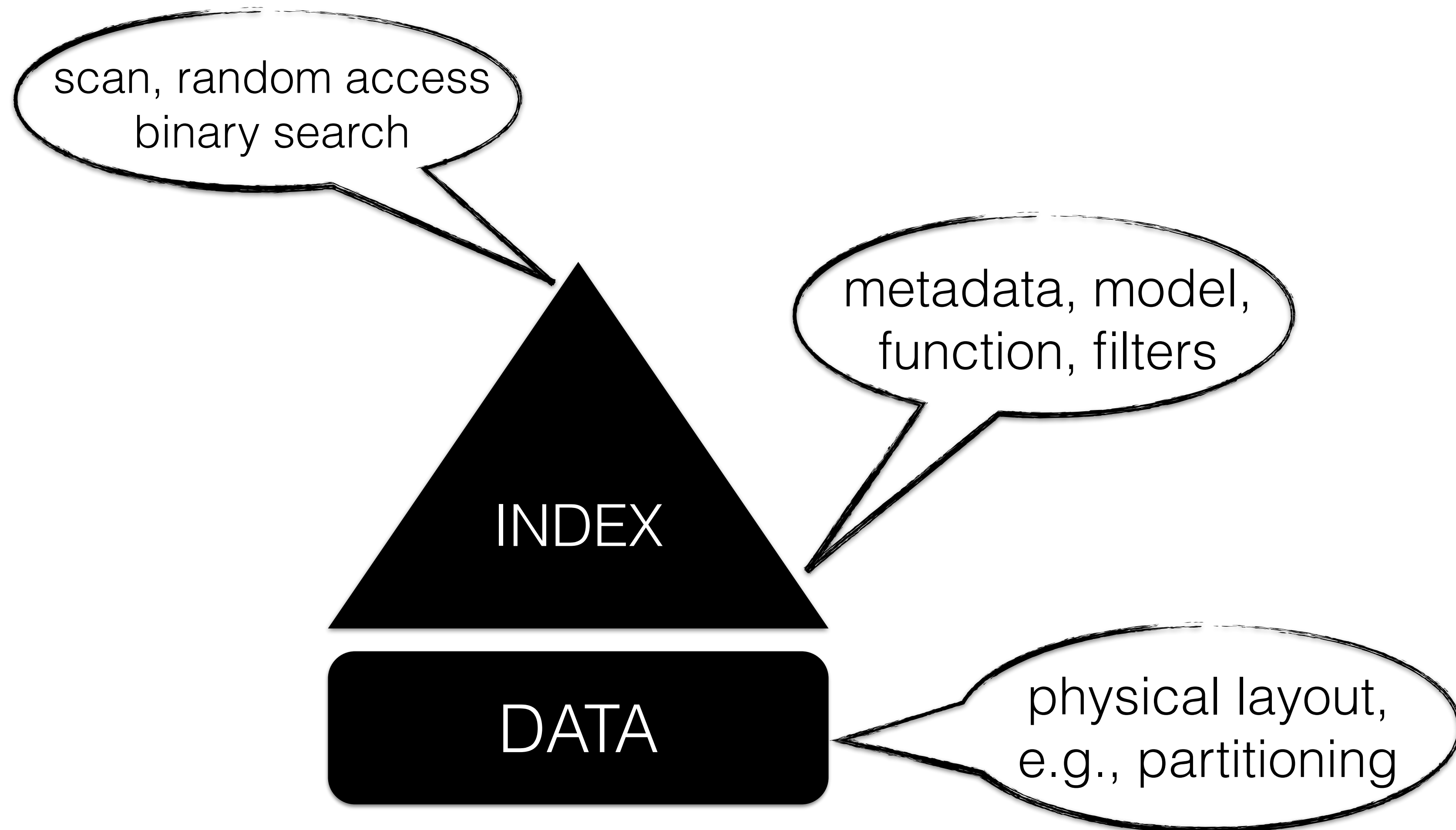


EVERY  
DESIGN:

**1** A **SET** OF  
CONCEPTS

**2** **EXISTING** OR  
**NEW** CONCEPTS

**3** ALL GOOD IDEAS IN THE 60s?



EVERY  
DESIGN:

**1** A **SET** OF  
CONCEPTS

**2** **EXISTING** OR  
**NEW** CONCEPTS

**3** ALL GOOD IDEAS IN THE 60s?



EVERY  
DESIGN:

**1** A **SET** OF  
CONCEPTS

**2** **EXISTING** OR  
**NEW** CONCEPTS

**3** ALL GOOD IDEAS IN THE 60s?

(ALMOST) ALL  
DESIGNS ARE A  
COMBINATION/  
TUNING  
OF **EXISTING**  
**CONCEPTS**





*action is for nothing  
hope the most holy  
am fear free form of  
ultimate I theory*

Nikos Kazantzakis, philosopher



Nikos Kazantzakis, philosopher

*action is  
the most holy  
ultimate form  
theory*

*I hope for nothing  
I fear nothing  
I am free*



Nikos Kazantzakis, philosopher

*action is  
the most holy  
ultimate form  
theory*

**NEW**

*I hope for nothing  
I fear nothing  
I am free*



Nikos Kazantzakis, philosopher

*action is  
the most holy  
ultimate form  
theory*

**NEW & BRILLIANT**

*I hope for nothing  
I fear nothing  
I am free*



milk + cream + sugar + vanilla/lemon

# **CEREAL MILK PANNA COTTA**

## **non obvious valid combinations**

*Christína Tosí*

**Best researchers: kids, young students, adults that stay kids**



milk + cream + sugar + vanilla/lemon

**CEREAL MILK PANNA COTTA**  
**non obvious valid combinations**

*Christina Tosi*

NP hard problem:  
2 PhD parents trying to get a **toddler** to wear **gloves**



NP hard problem:  
2 PhD parents trying to get a **toddler** to wear **gloves**



NP hard problem:  
2 PhD parents trying to get a **toddler** to wear **gloves**

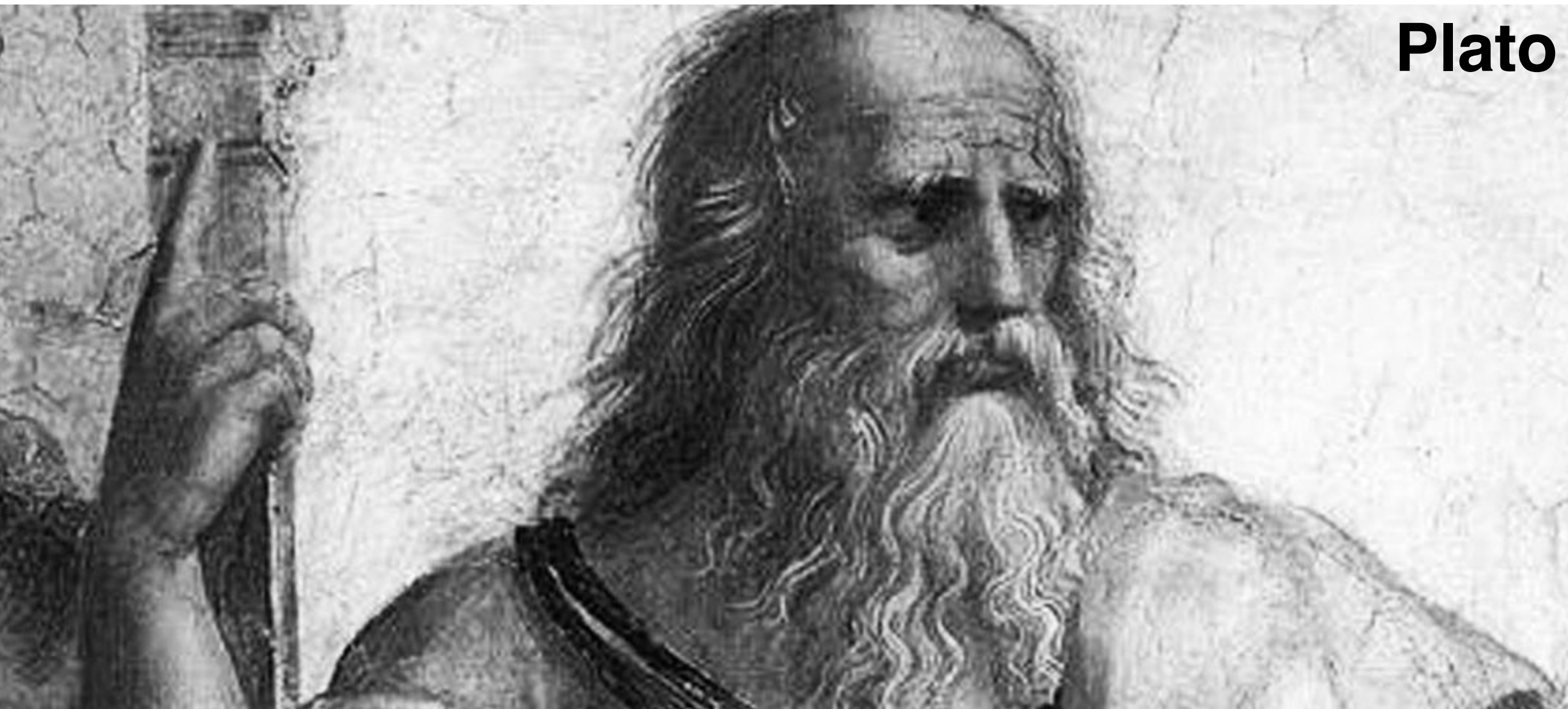
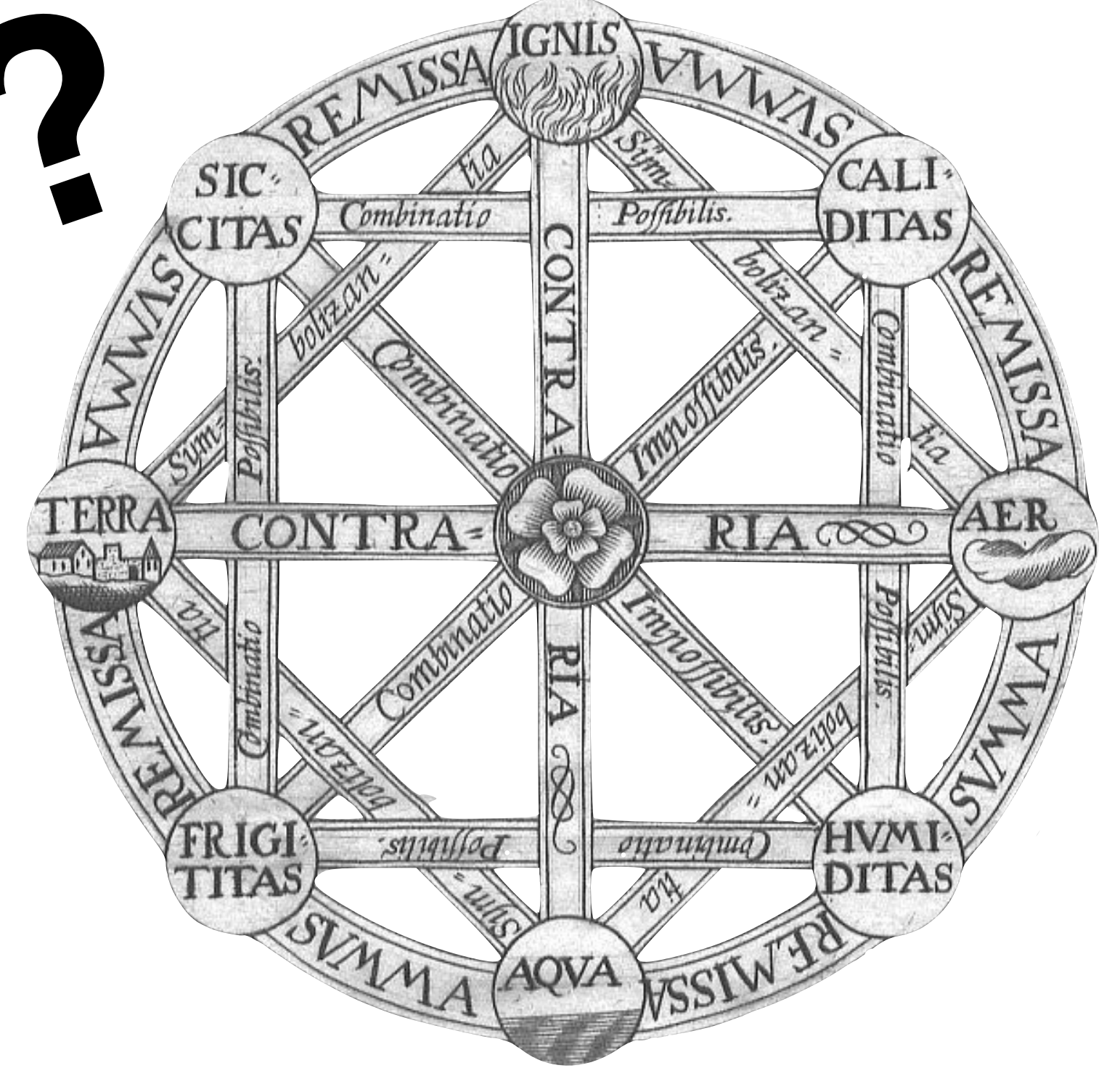






socks!

# what is creativity?

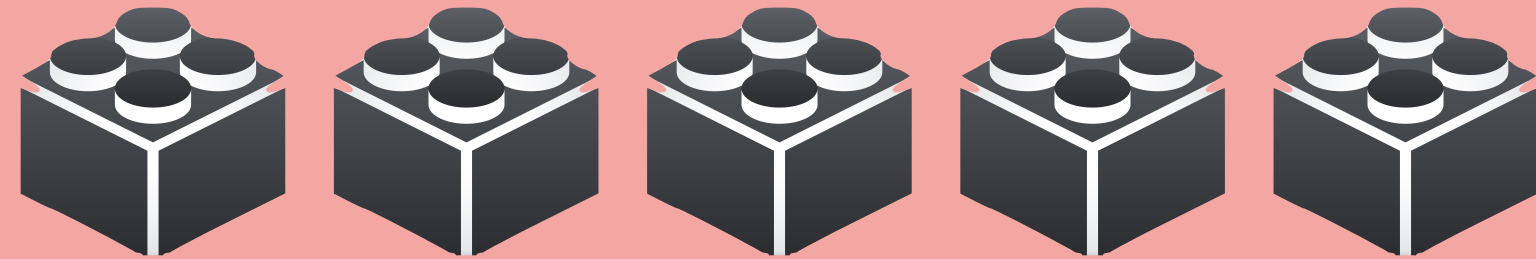


Plato

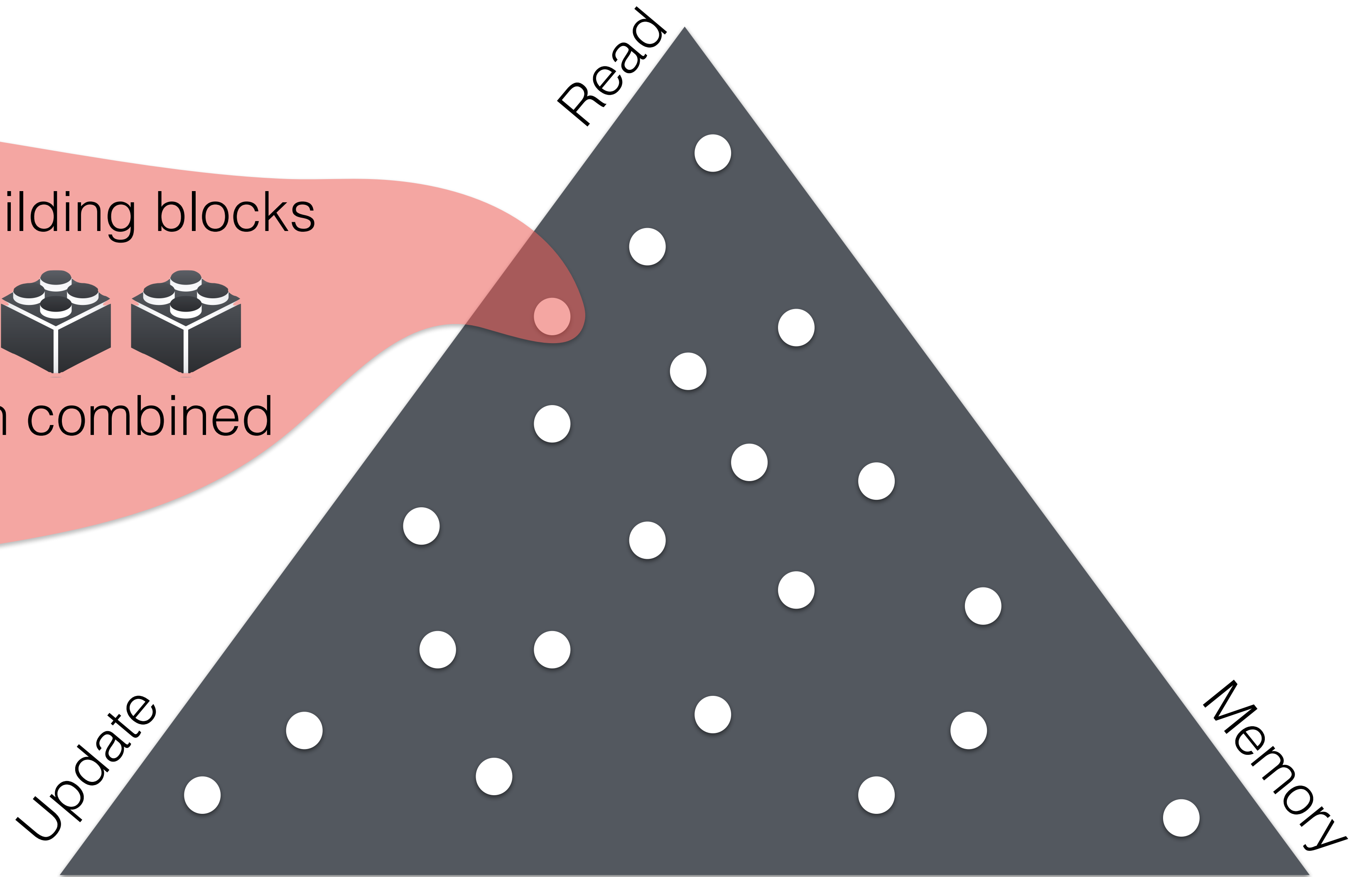


Leibniz

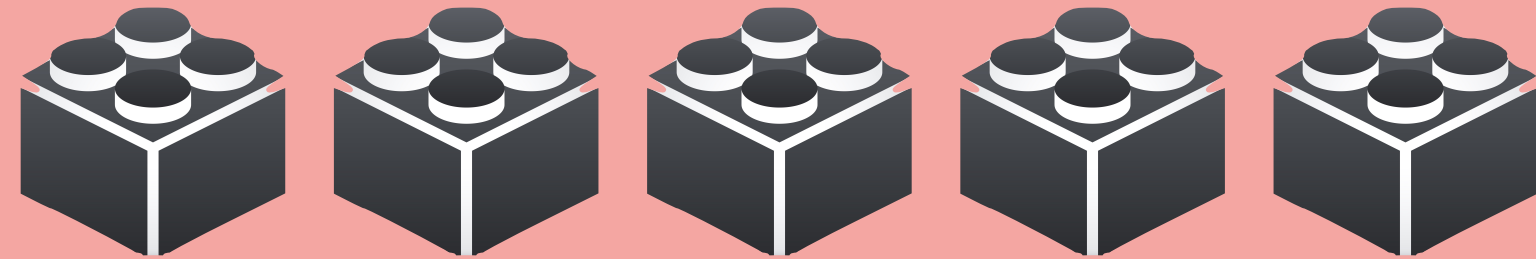
**fundamental** building blocks



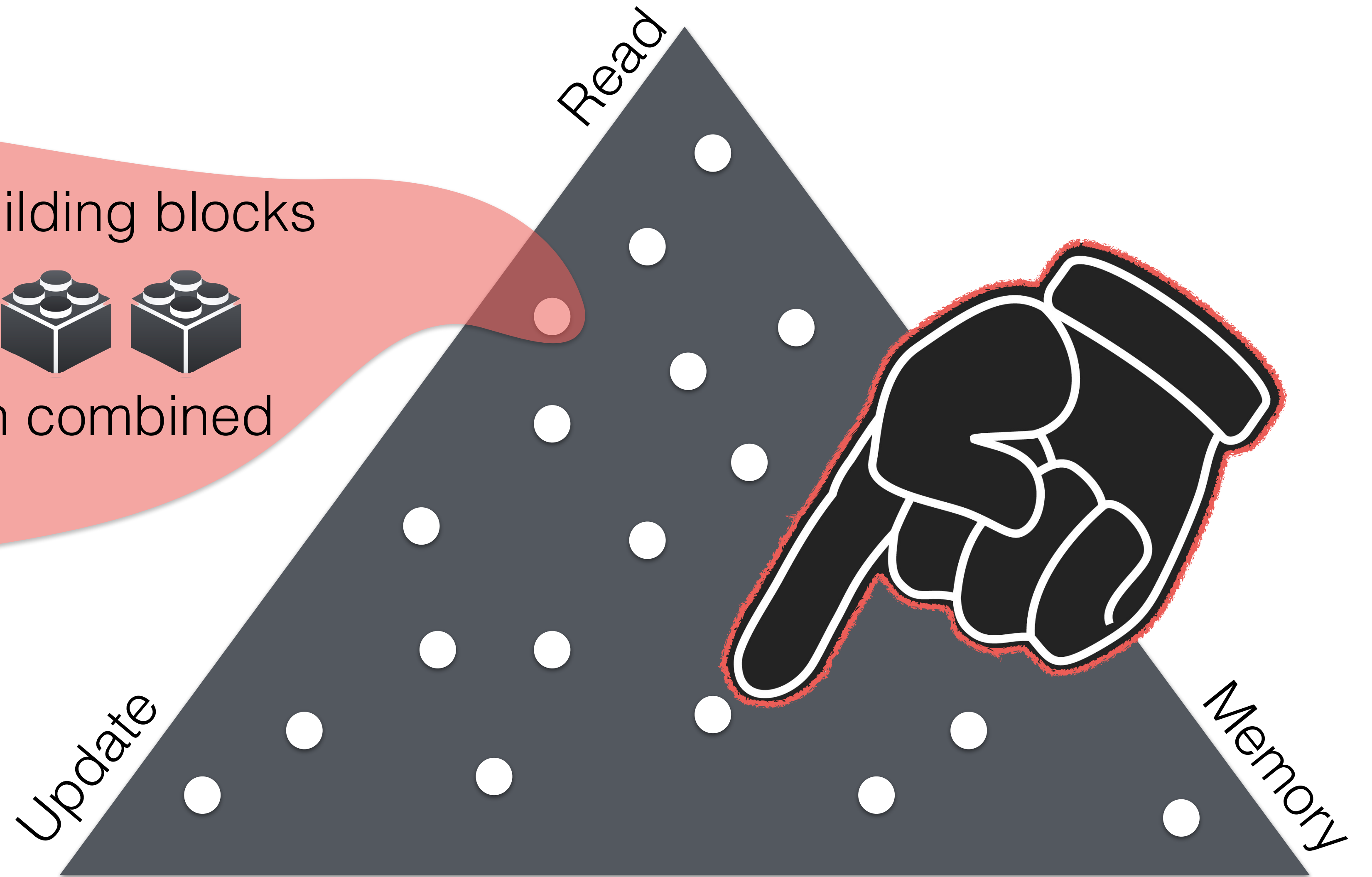
**properties** when combined



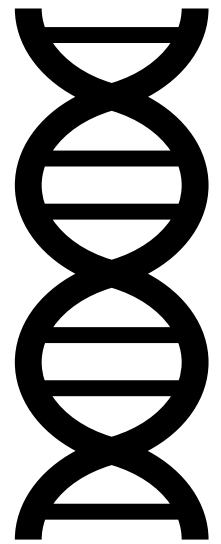
**fundamental** building blocks



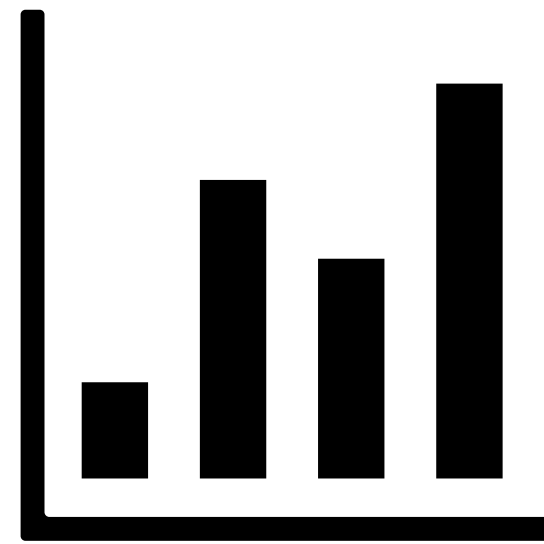
**properties** when combined



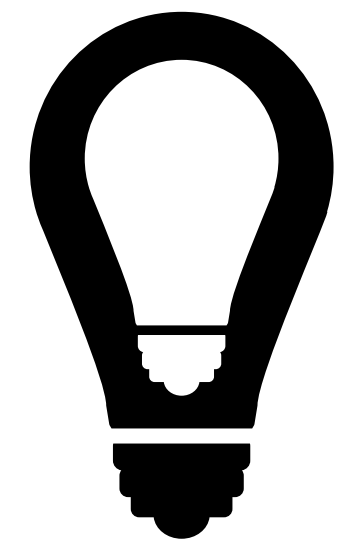
# Three steps required



DESIGN SPACE



COST SYNTHESIS



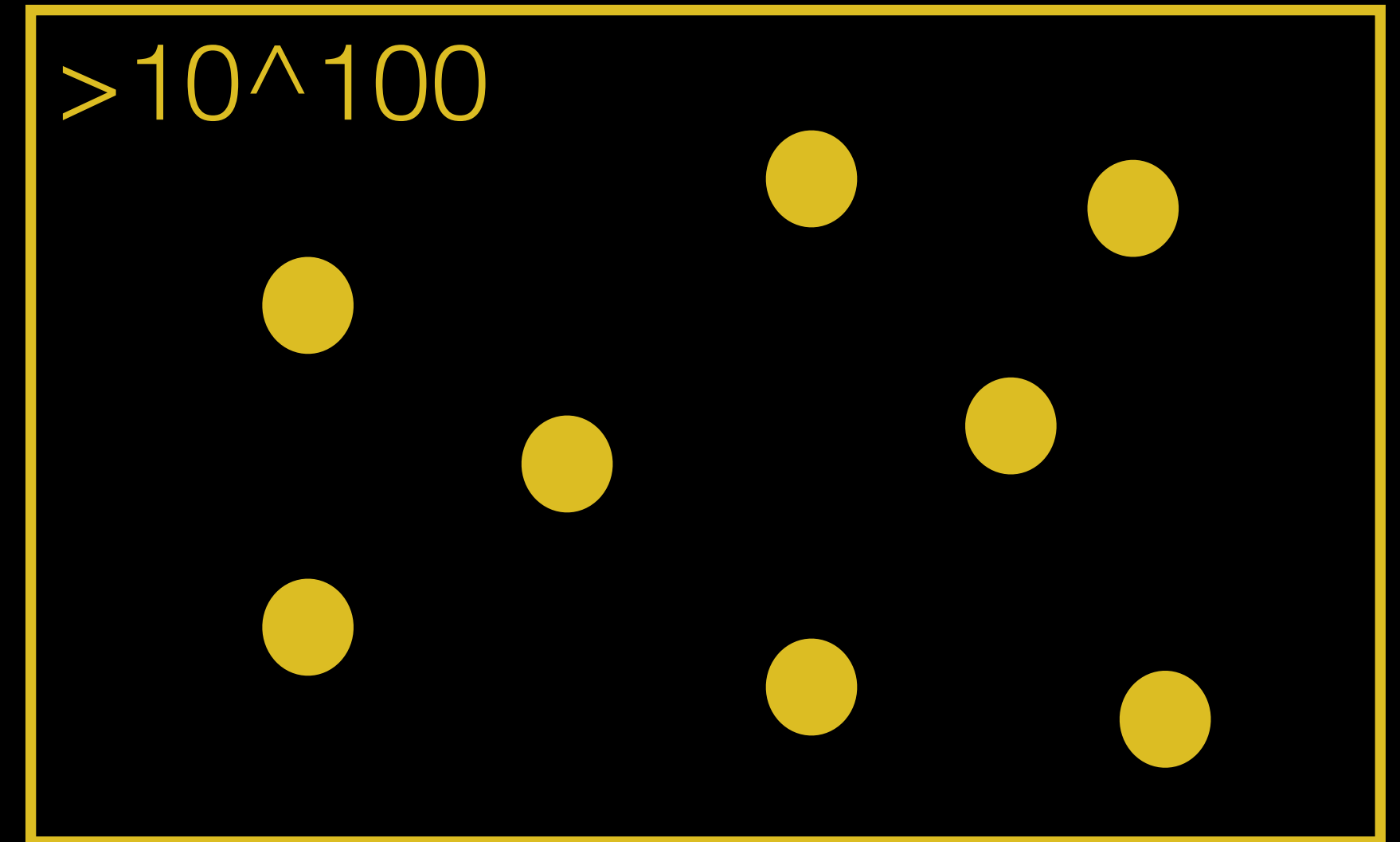
WHAT-IF

# 1. DESIGN SPACE

data layout of data structures

algorithm design

systems: interactions of components

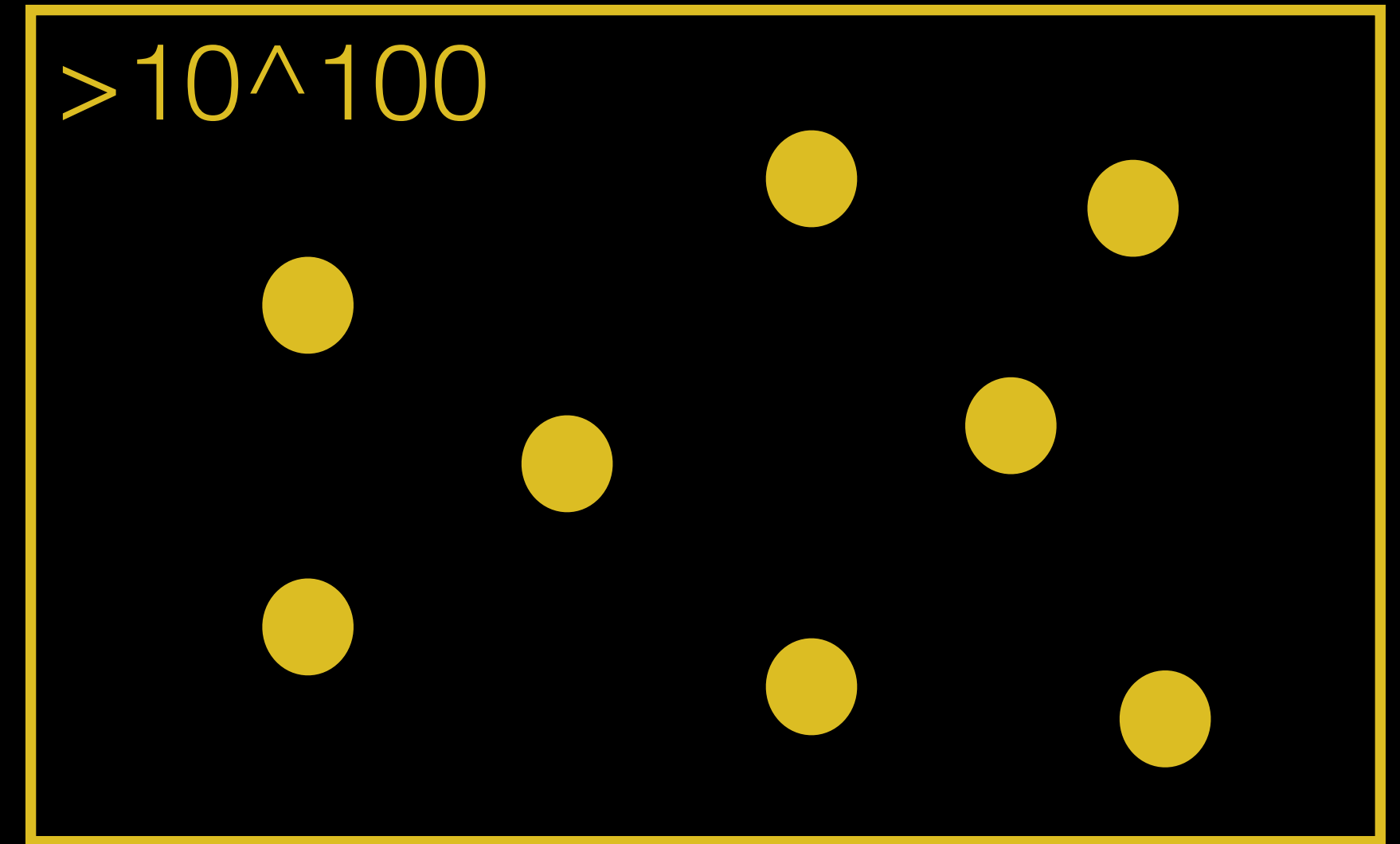


## 1. DESIGN SPACE

data layout of data structures

algorithm design

systems: interactions of components



## 2. NAVIGATE SEARCH SPACE

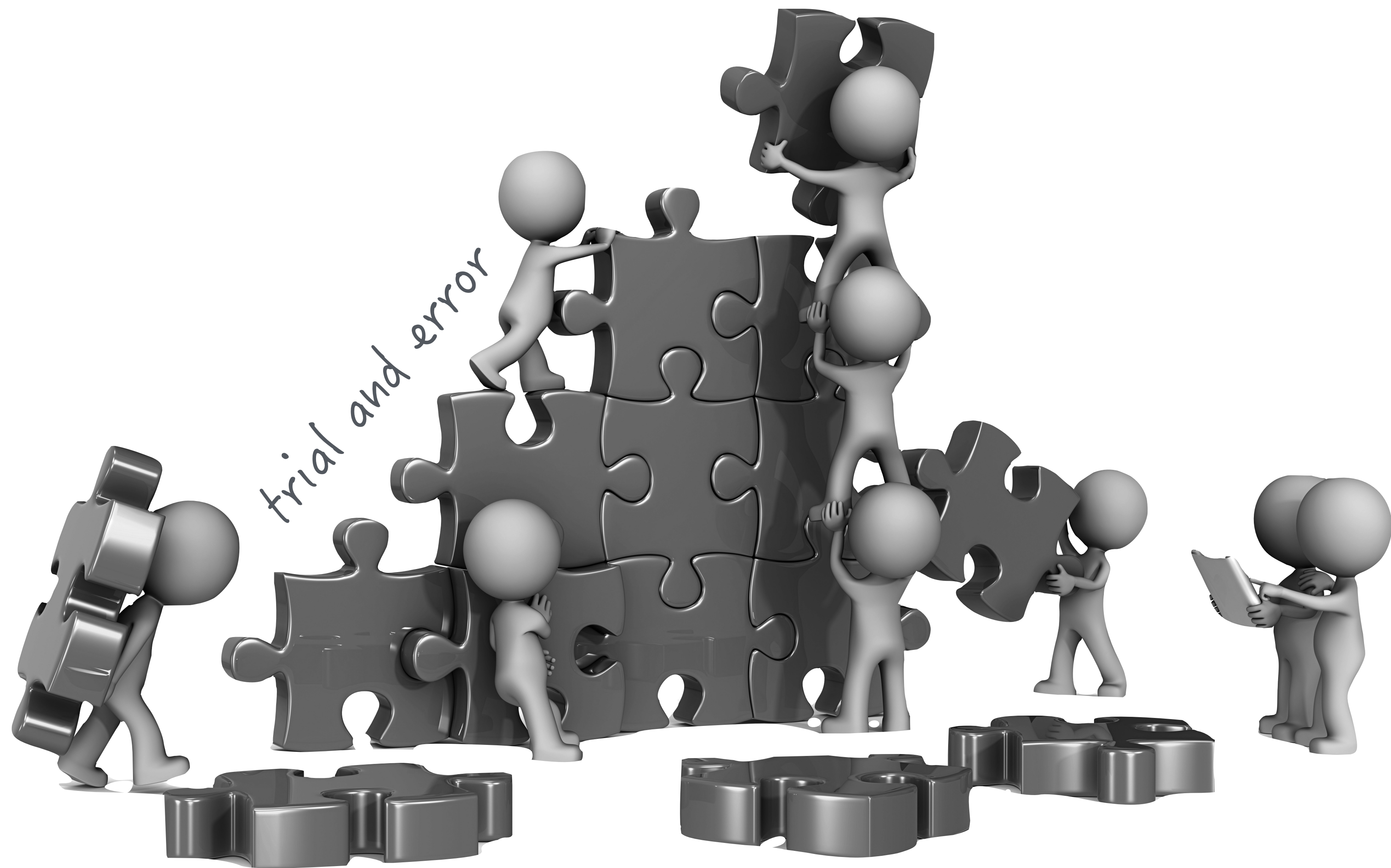
cost synthesis: computation and data movement

learned cost models in memory/parallelism

design continuums to shrink space

# **Step 1:**

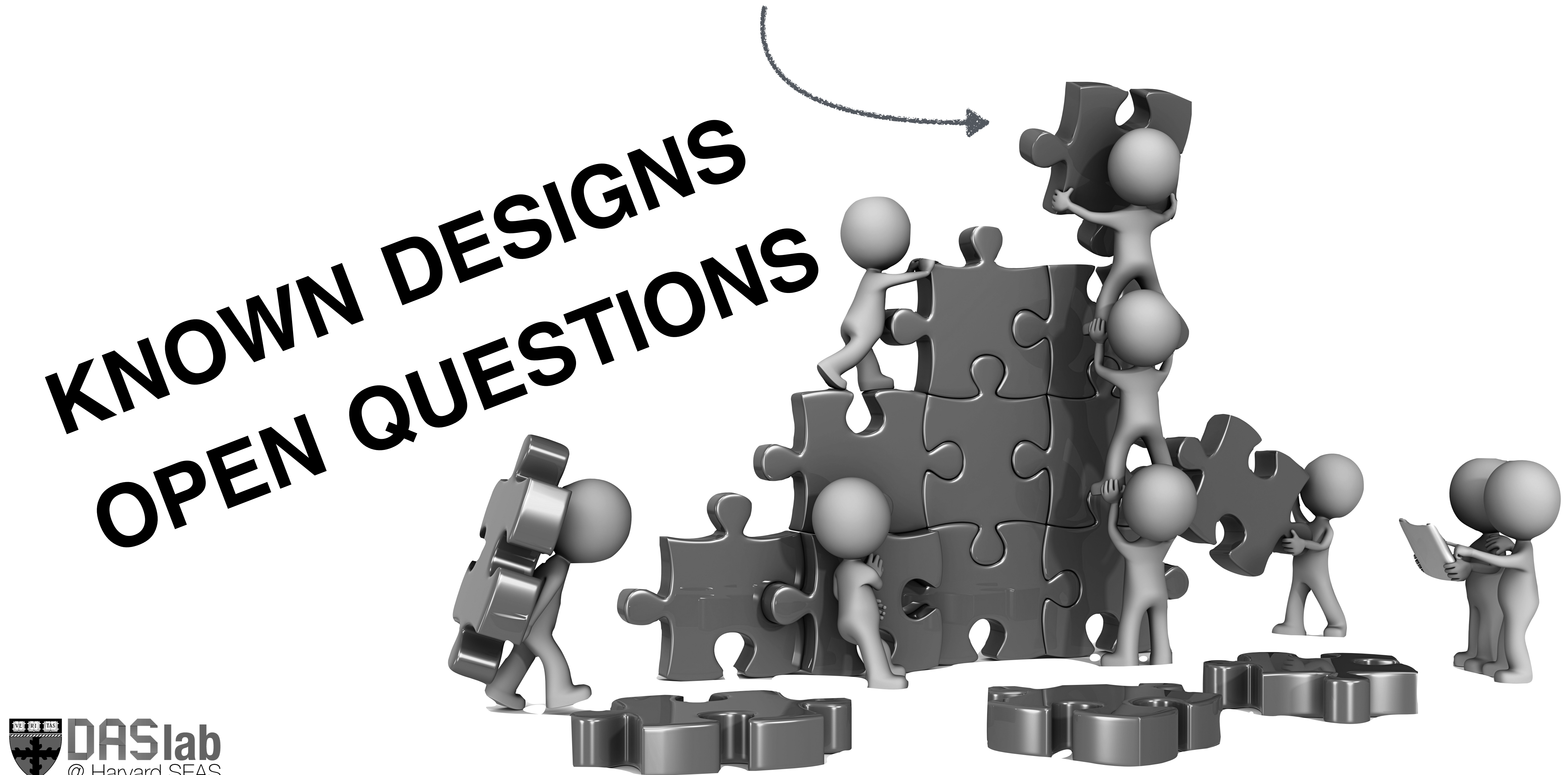
## **First Principles of Design to Construct Design Space**



FIRST PRINCIPLE: design concept that does not break further



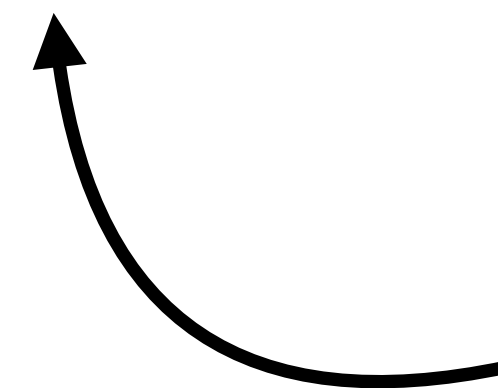
FIRST PRINCIPLE: design concept that does not break further



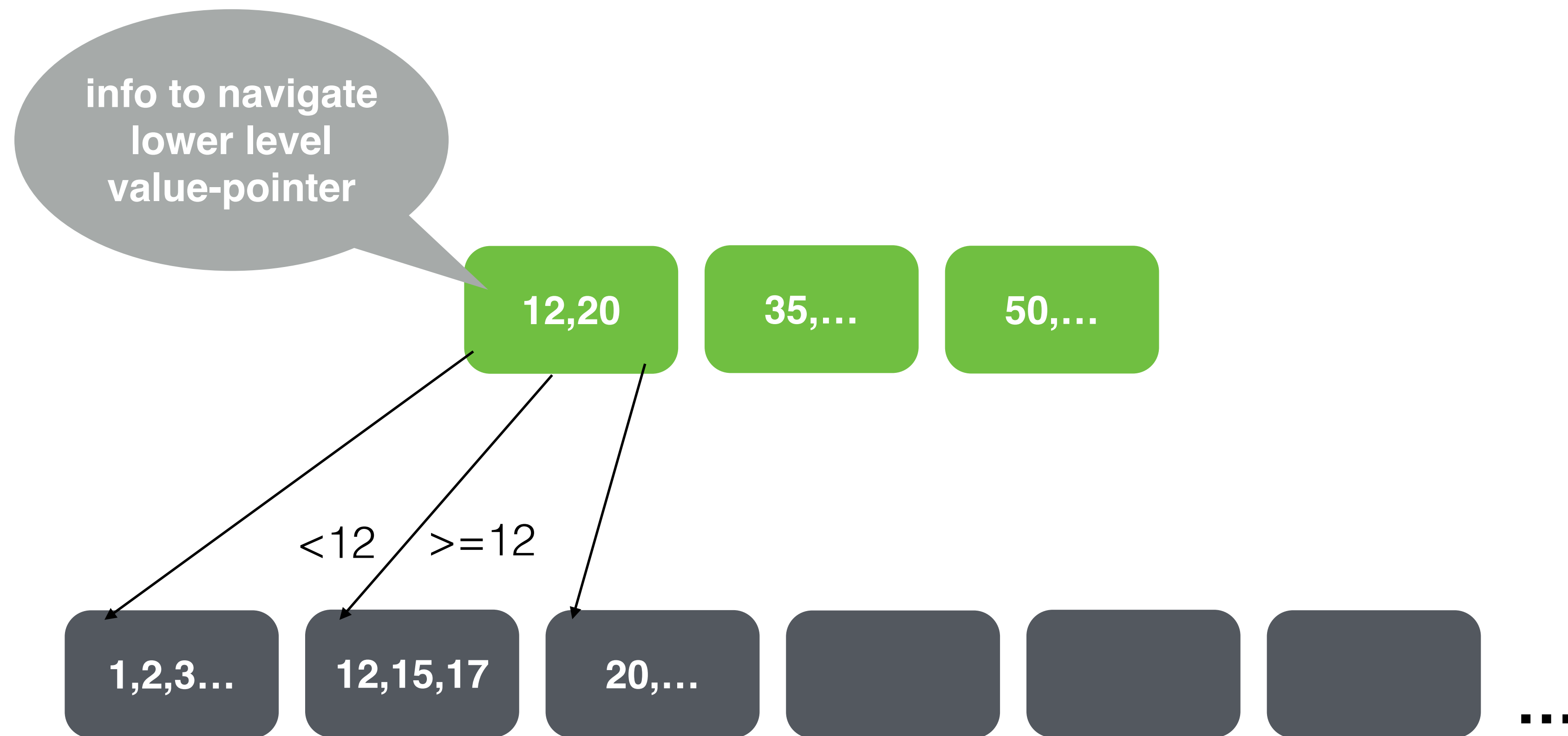
**To think about design principles/design space  
we need to first fully understand an area in extreme detail**

**To think about design principles/design space  
we need to first fully understand an area in extreme detail**

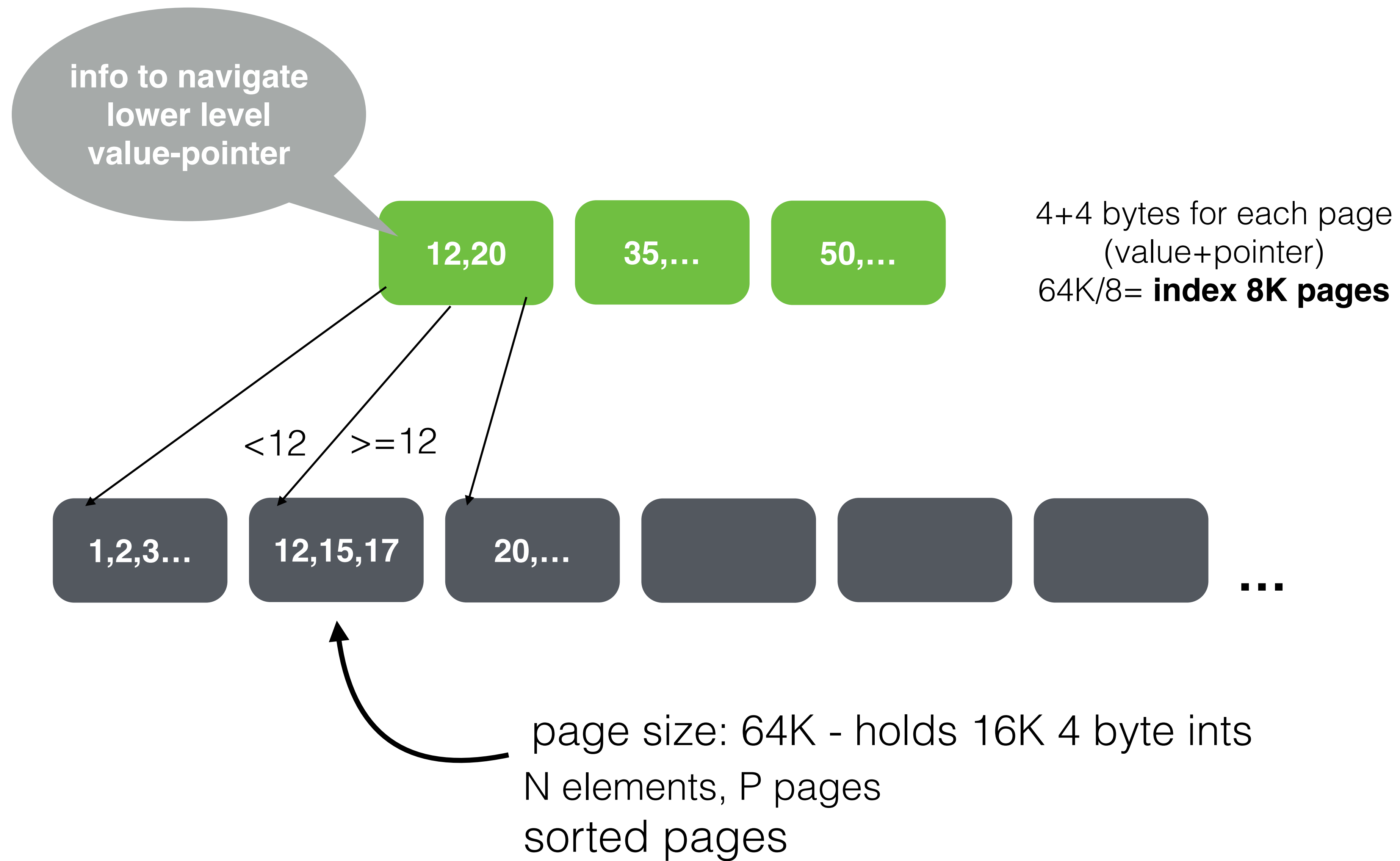
next: 2 data structures that drive 90% of modern kv-stores  
B-tree & LSM-tree

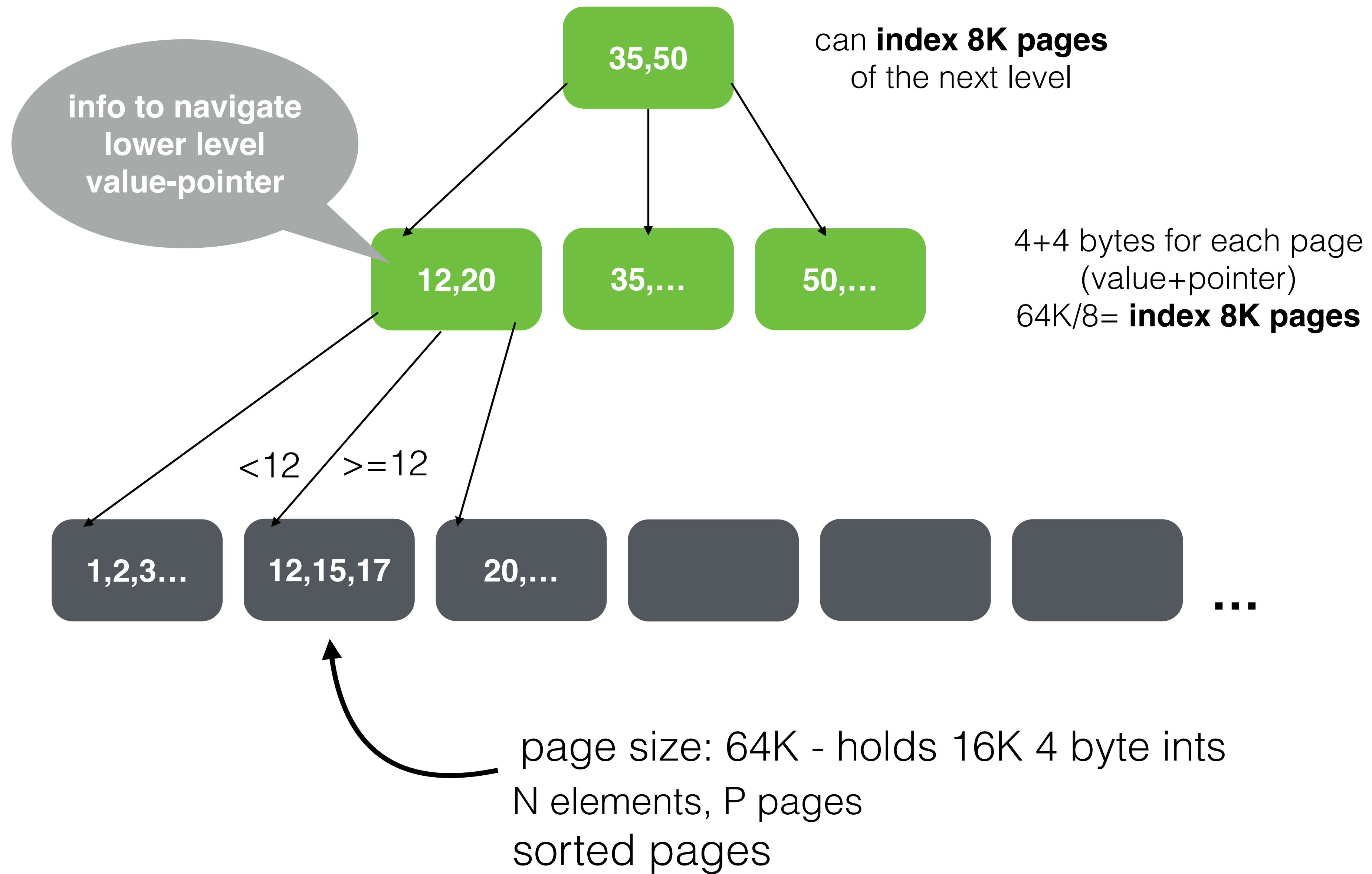


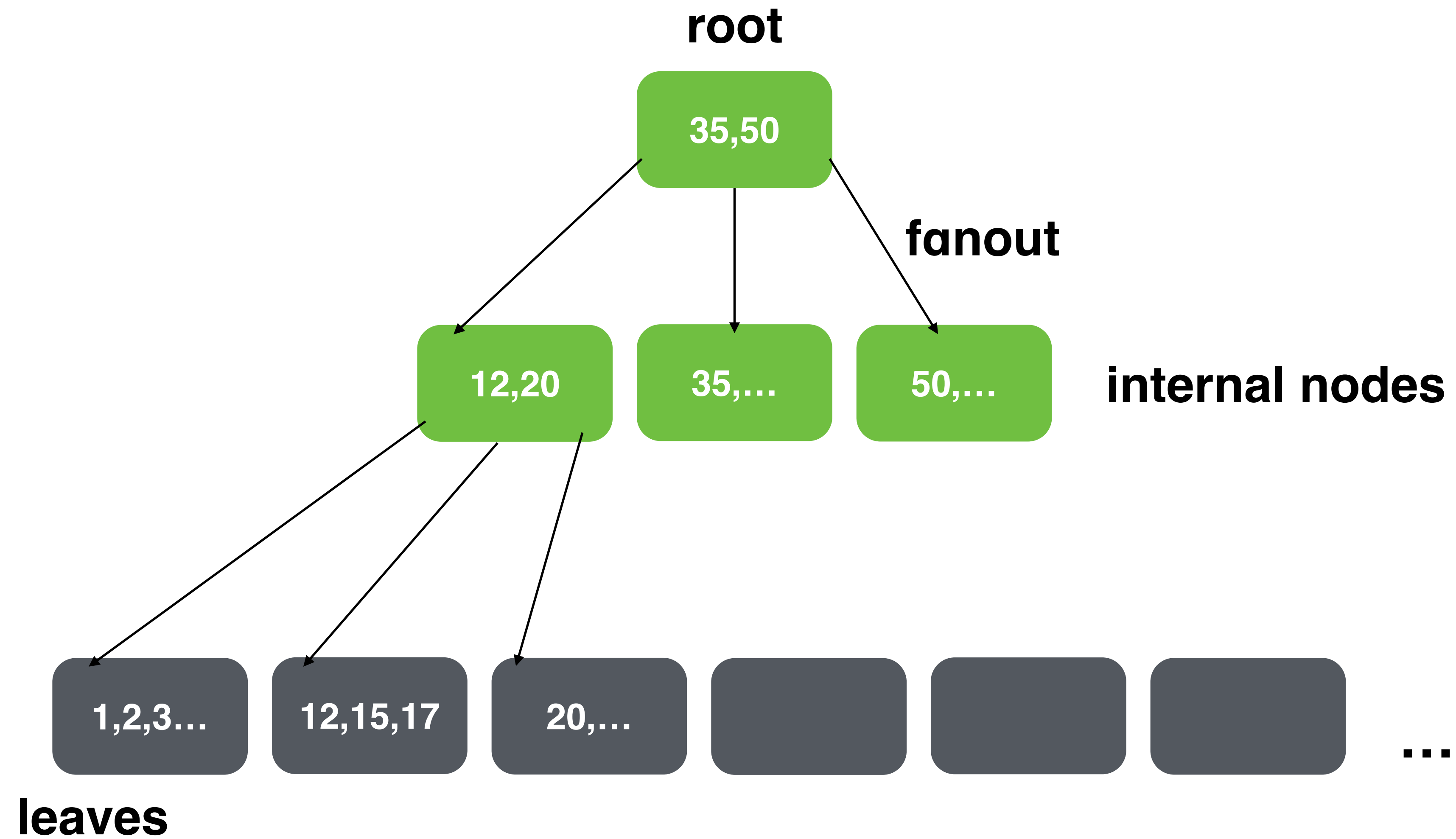
page size: 64K - holds 16K 4 byte ints  
N elements, P pages  
sorted pages

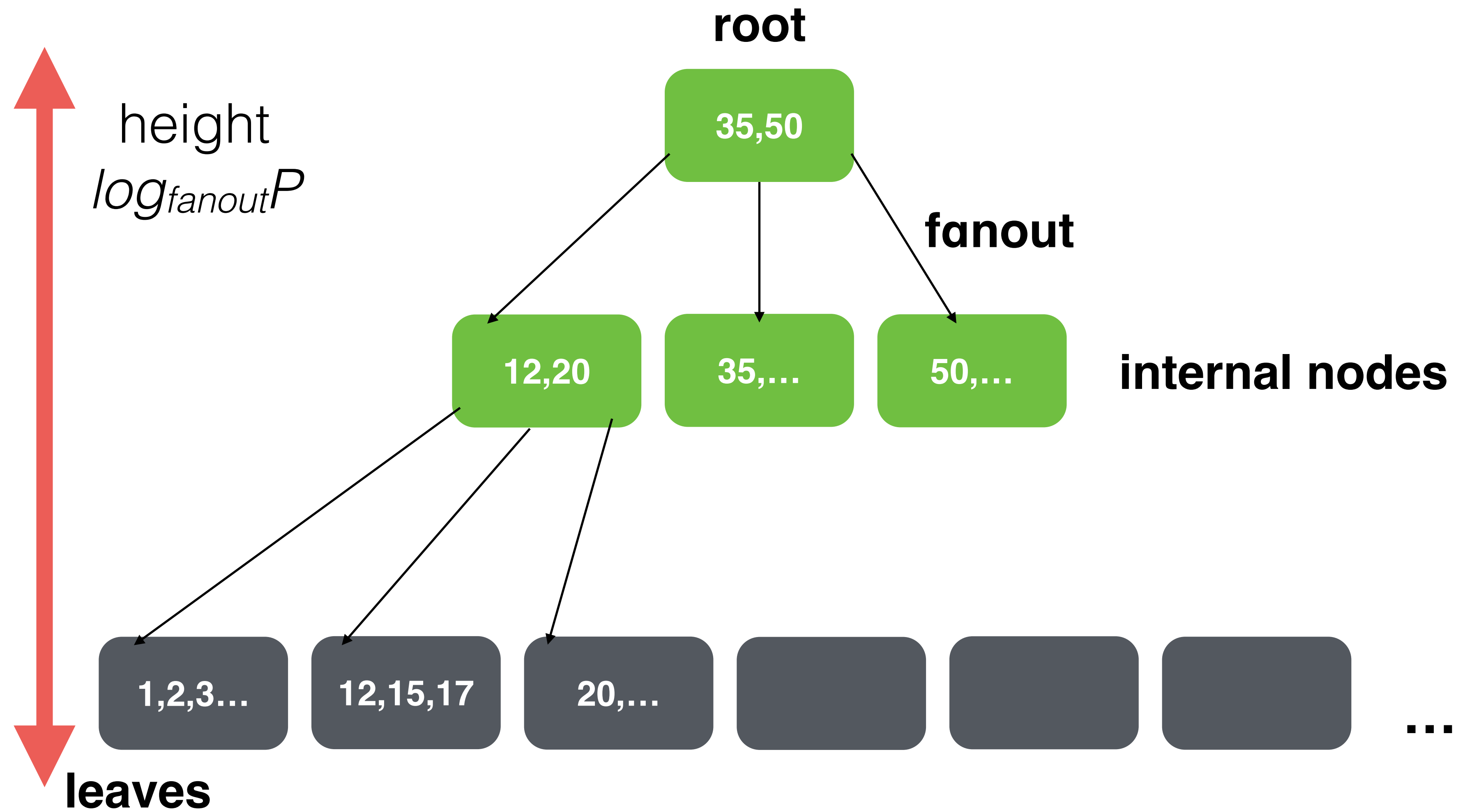


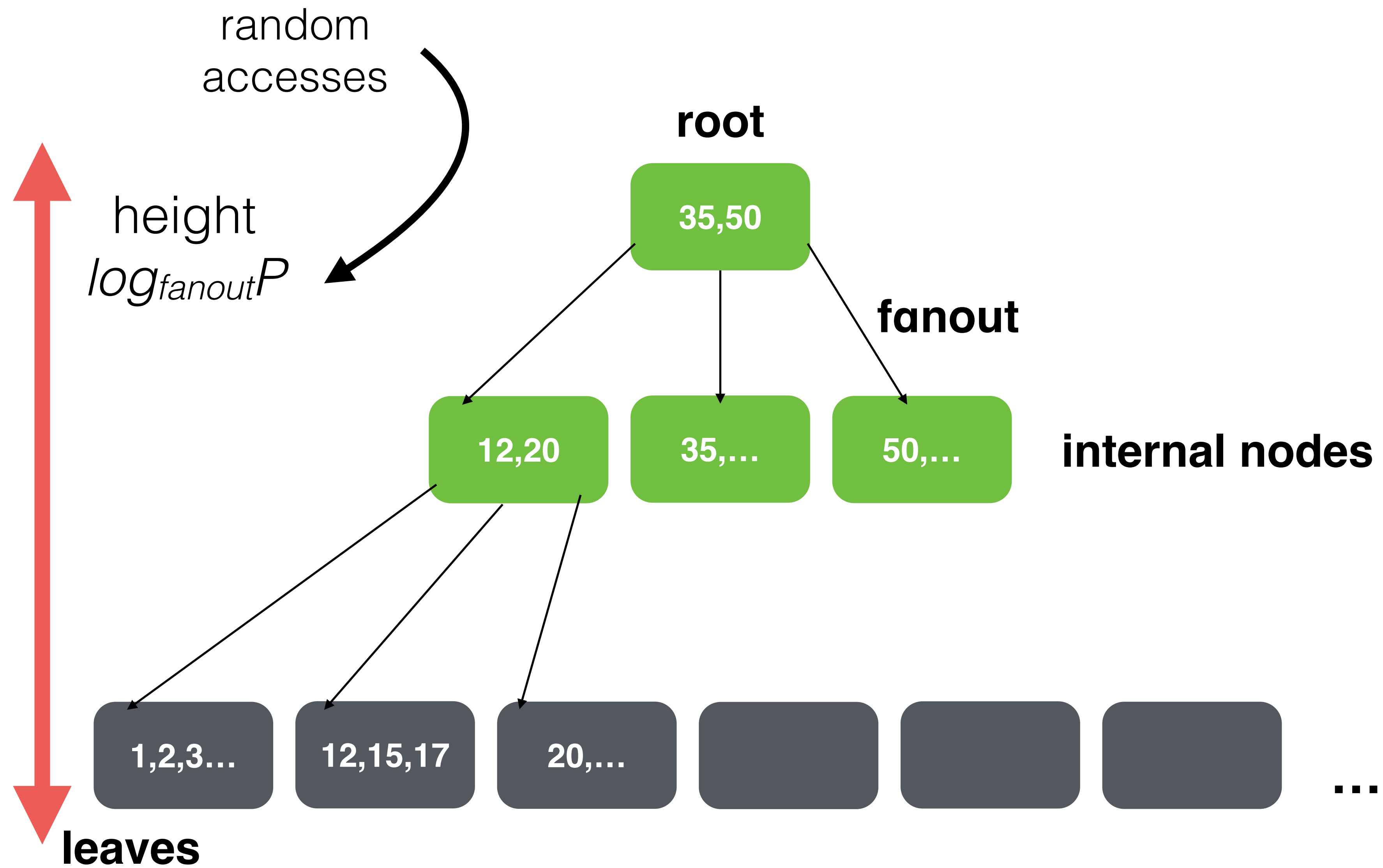
page size: 64K - holds 16K 4 byte ints  
N elements, P pages  
sorted pages

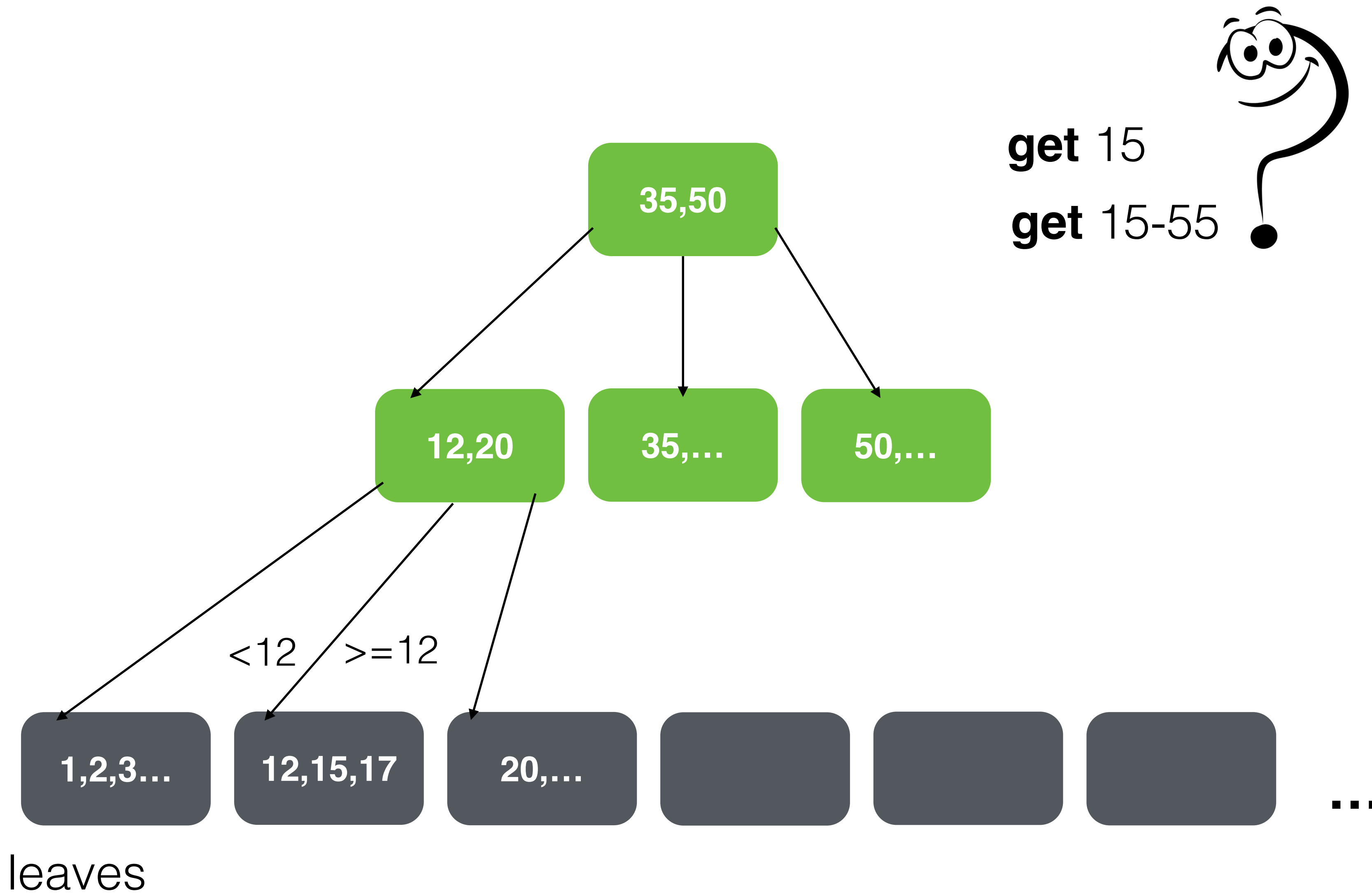


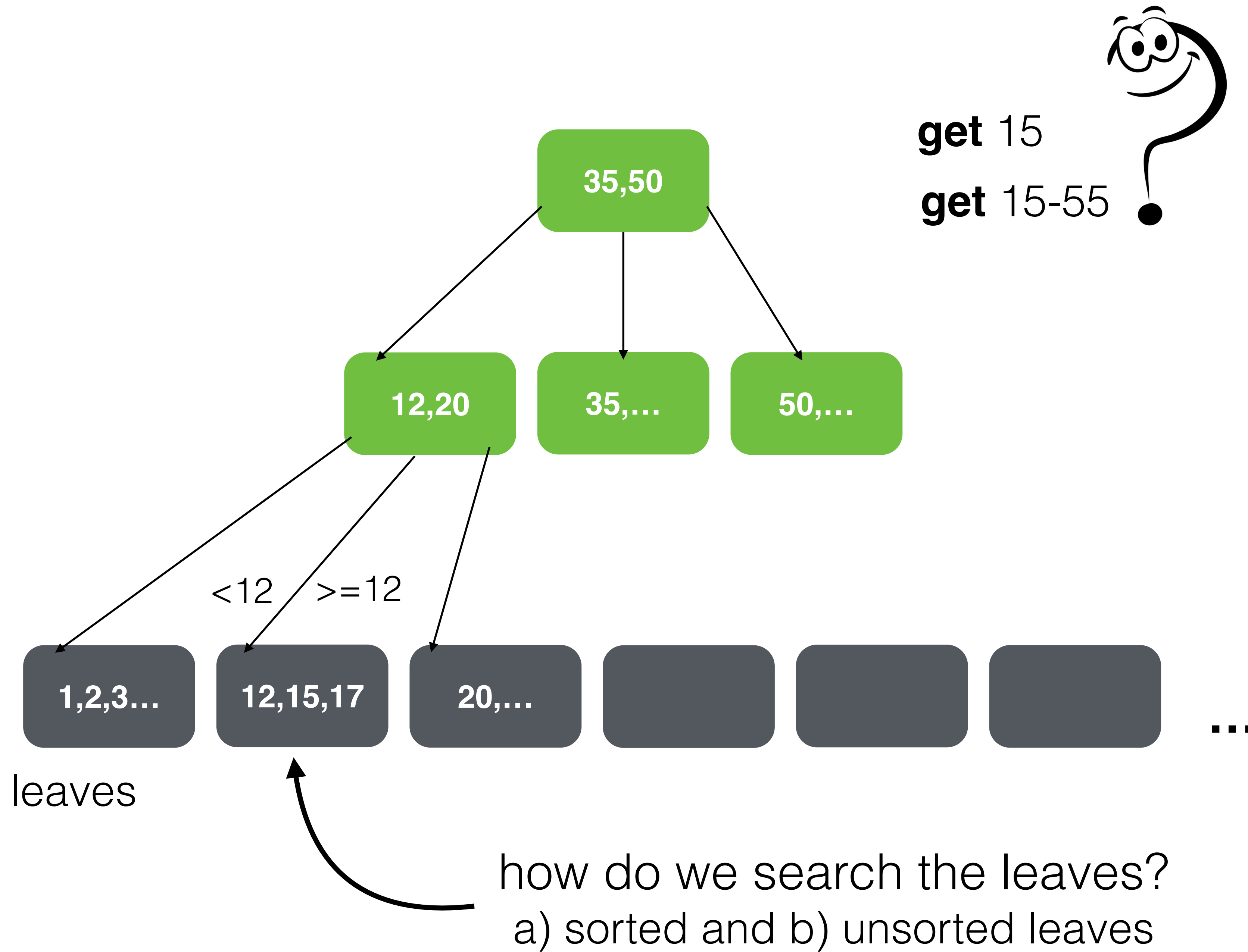








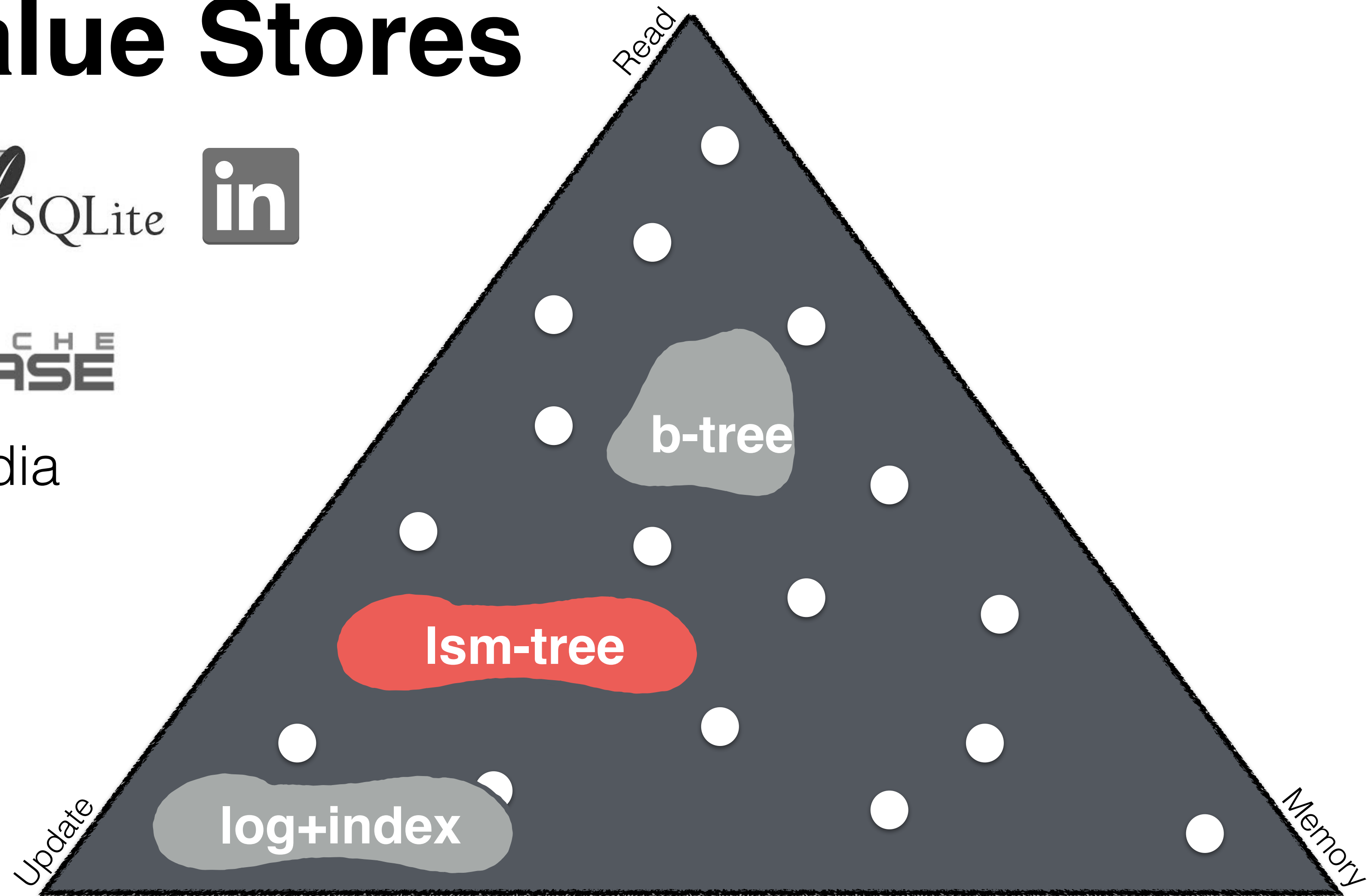




# NoSQL Key-value Stores



machine learning   social media  
smart homes   web browsers  
phones   web-based apps  
security   health devices  
graphs   analytics



**insert (key-value)**

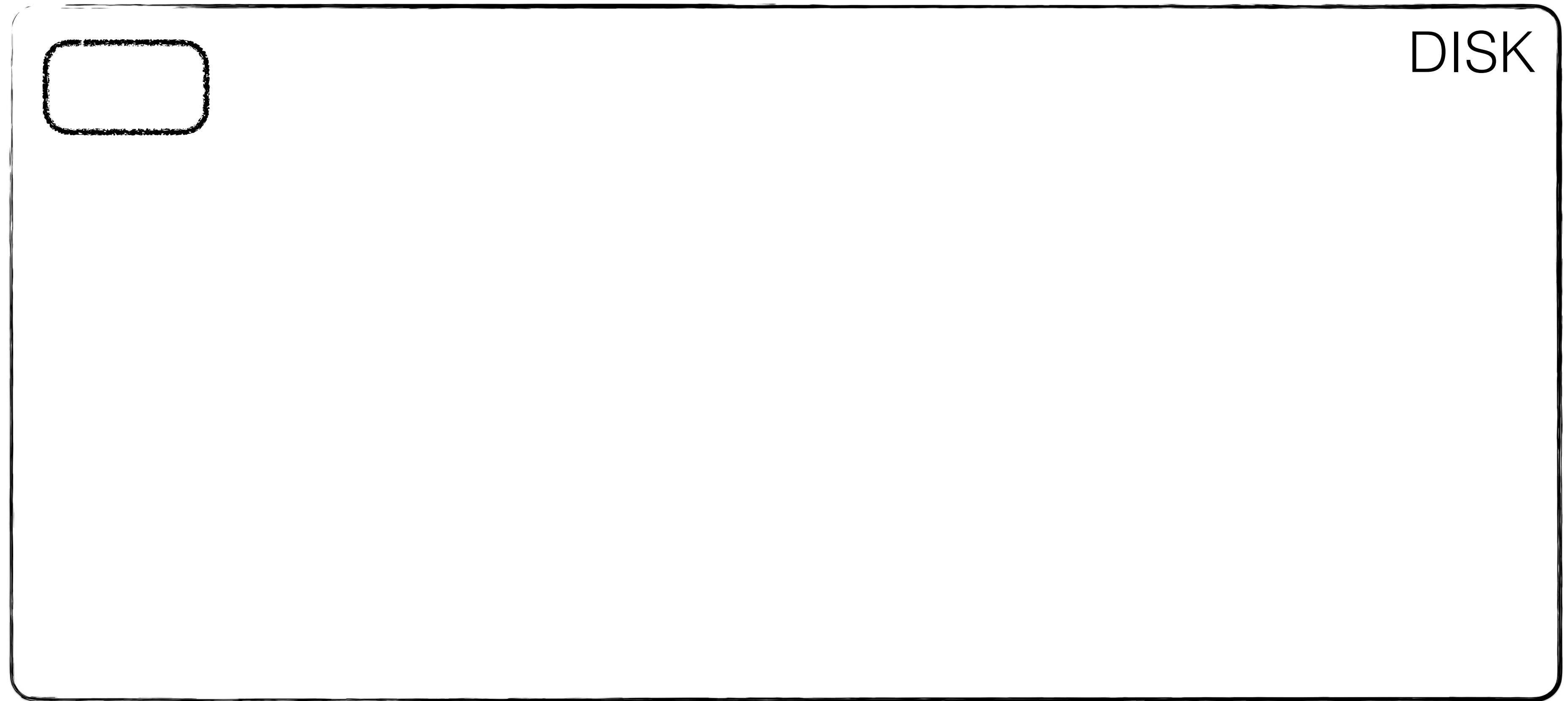


buffer

MEMORY

DISK

MEMORY  
DISK



MEMORY  
DISK

Level 1

**insert (key-value)**



buffer

MEMORY

DISK

Level 1

MEMORY  
DISK

Level 1

MEMORY  
DISK

Level 1

**insert (key-value)**



buffer

MEMORY

DISK

Level 1

MEMORY  
DISK



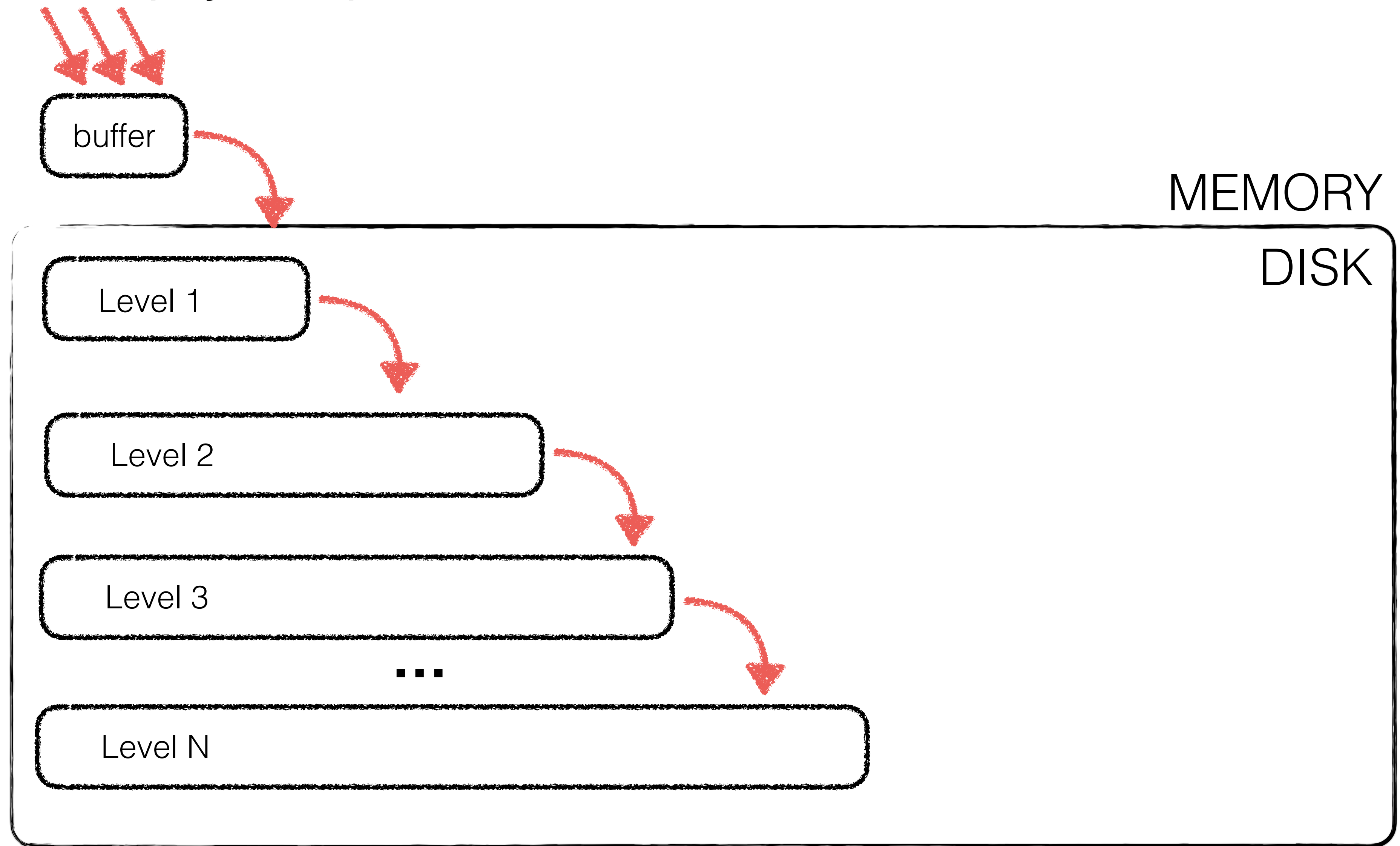
Level 2

MEMORY  
DISK

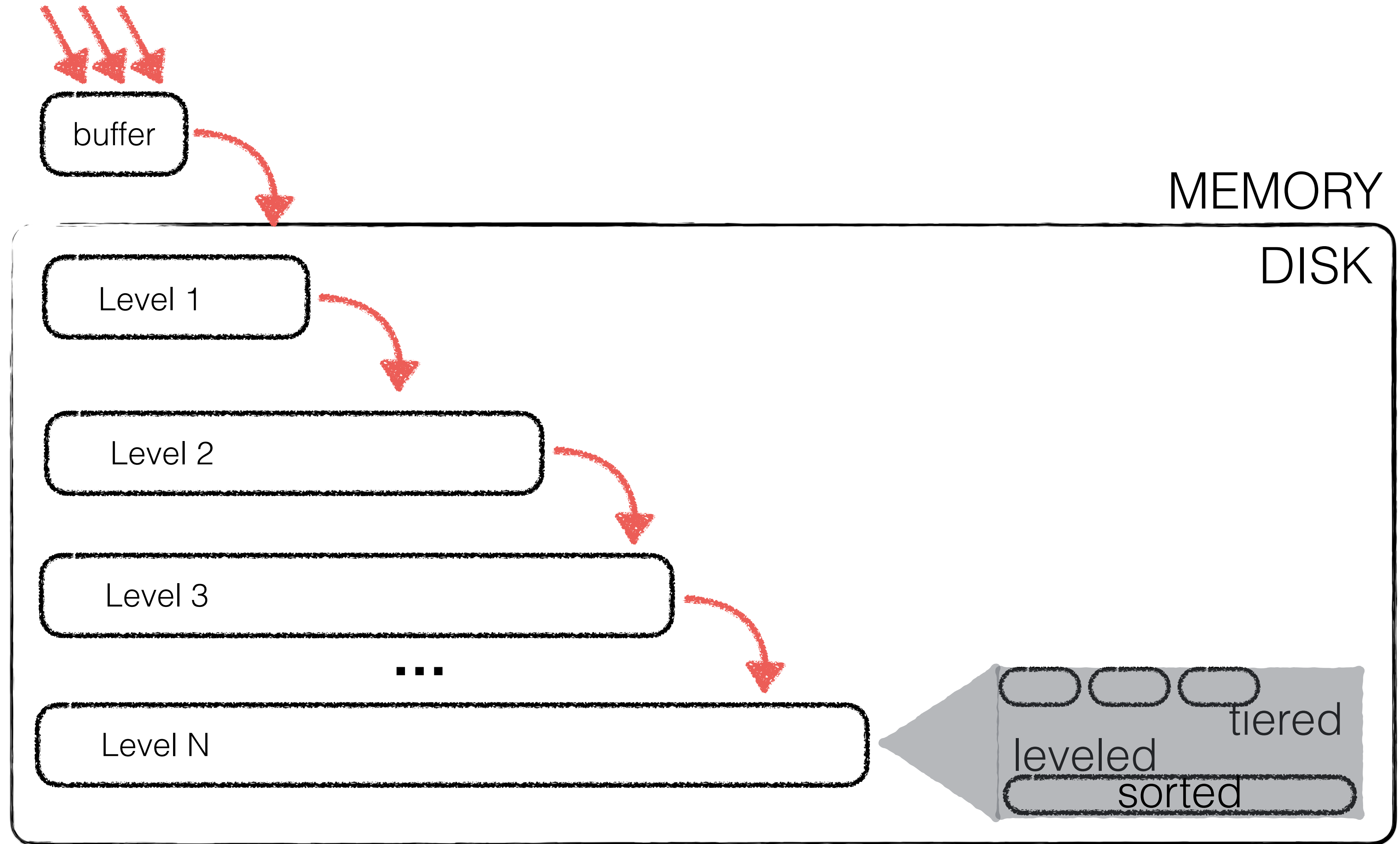
Level 1

Level 2

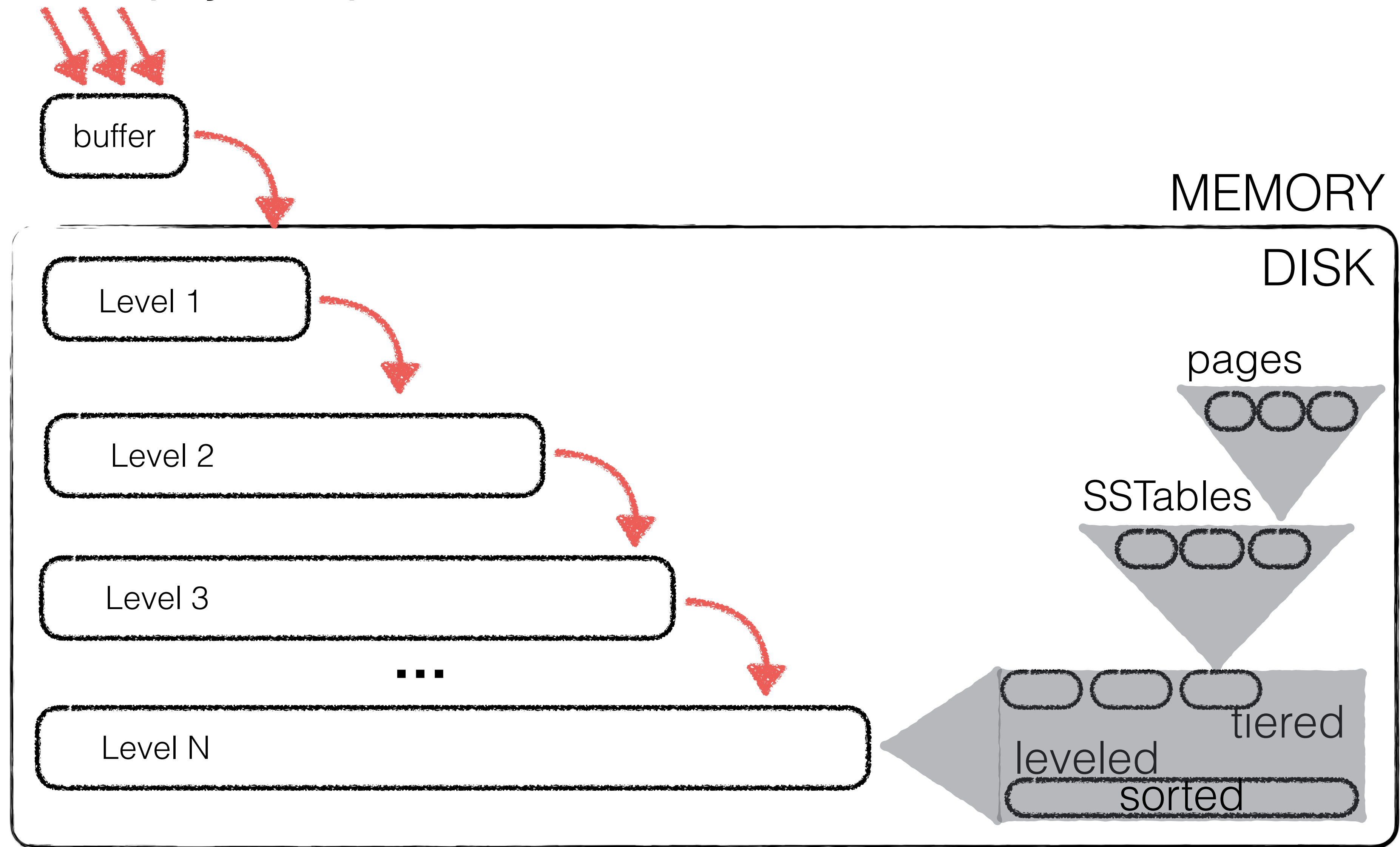
**insert (key-value)**



**insert (key-value)**

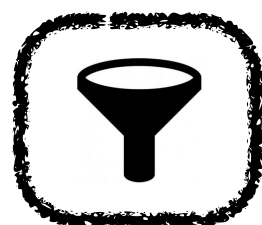


## insert (key-value)



[1,0,0,1,1,1]  
hash fun.

bloom  
filters

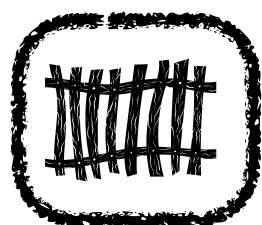
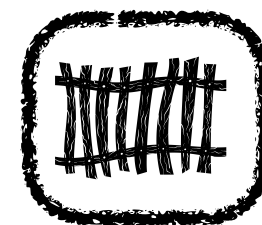
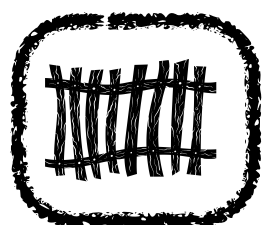


...

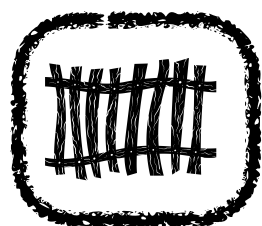


[min-max]  
/page

fence  
pointers



...



buffer

Level 1

Level 2

Level 3

...

Level N

MEMORY

DISK

pages



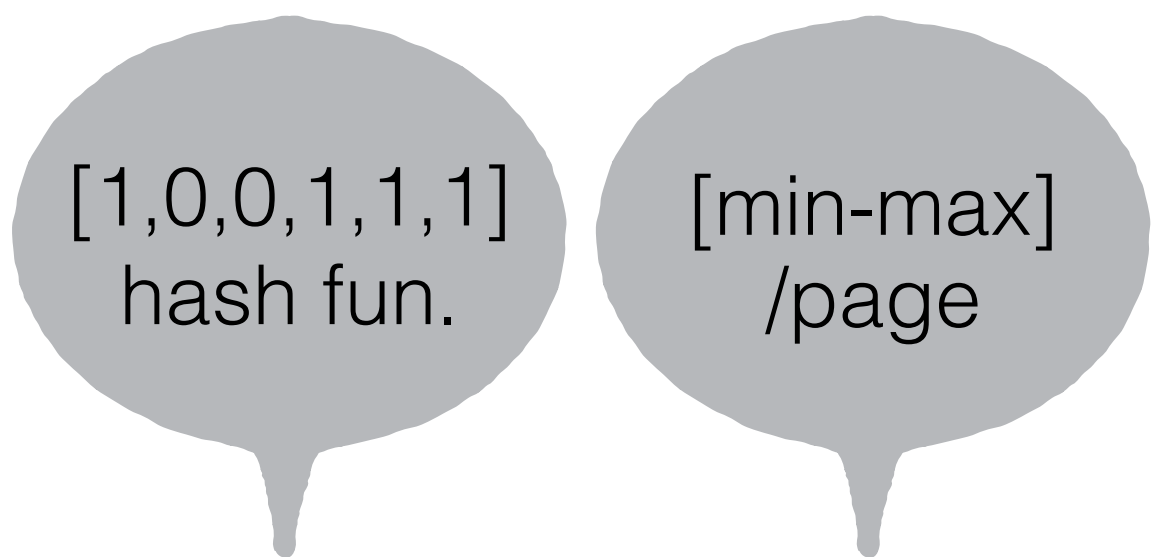
SSTables



tiered

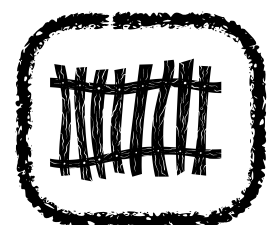
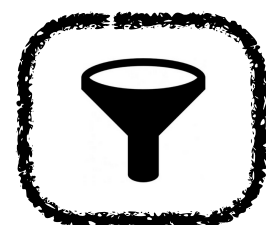
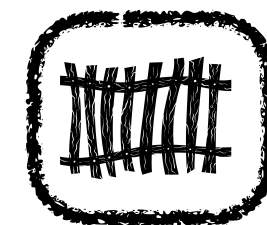
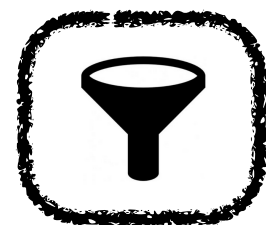
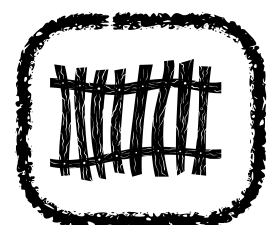
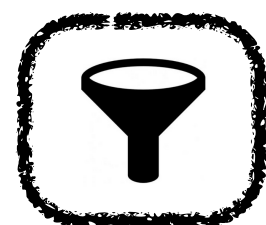
leveled

sorted



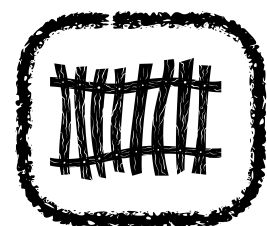
bloom  
filters

fence  
pointers



...

...



get (key)

buffer

Level 1

Level 2

Level 3

...

Level N

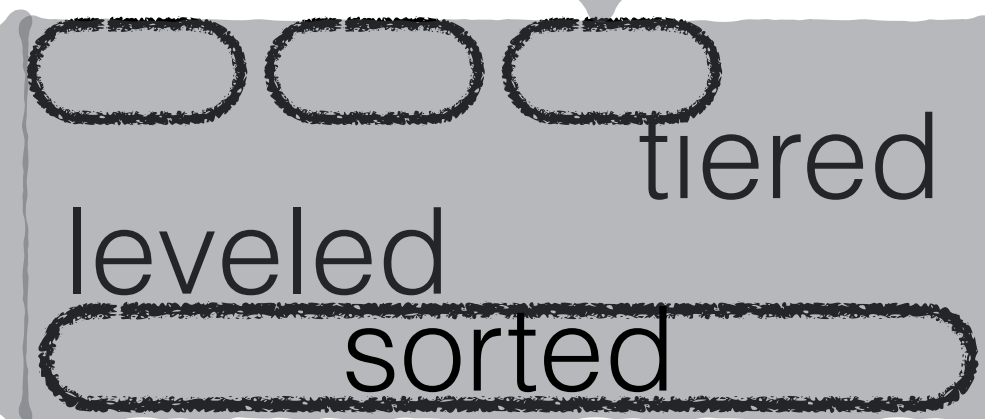
MEMORY

DISK

pages



SSTables



[1,0,0,1,1,1]  
hash fun.

[min-max]  
/page

bloom  
filters

fence  
pointers

get (key)

buffer

MEMORY

DISK

pages

SSTables

tiered

leveled

sorted

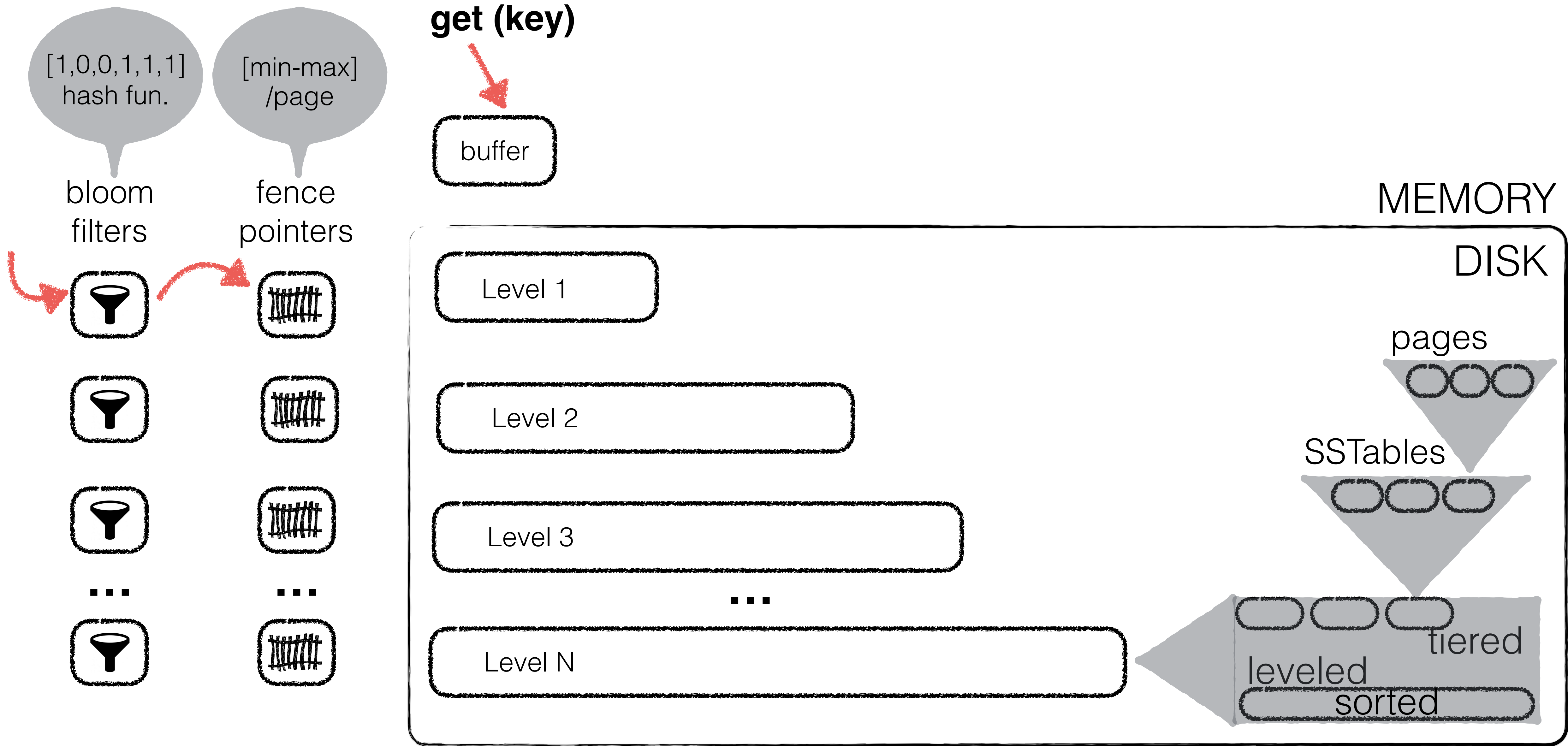
Level 1

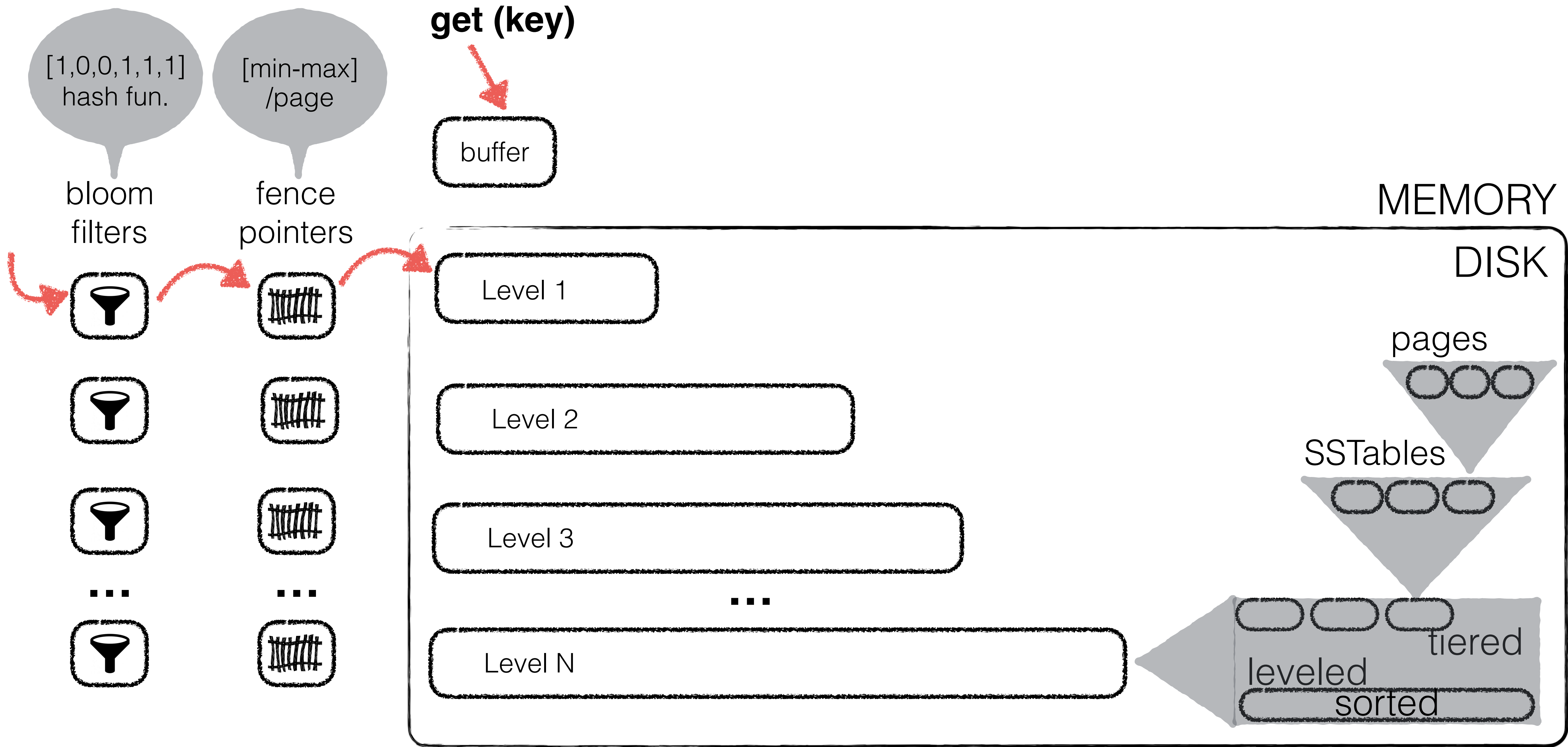
Level 2

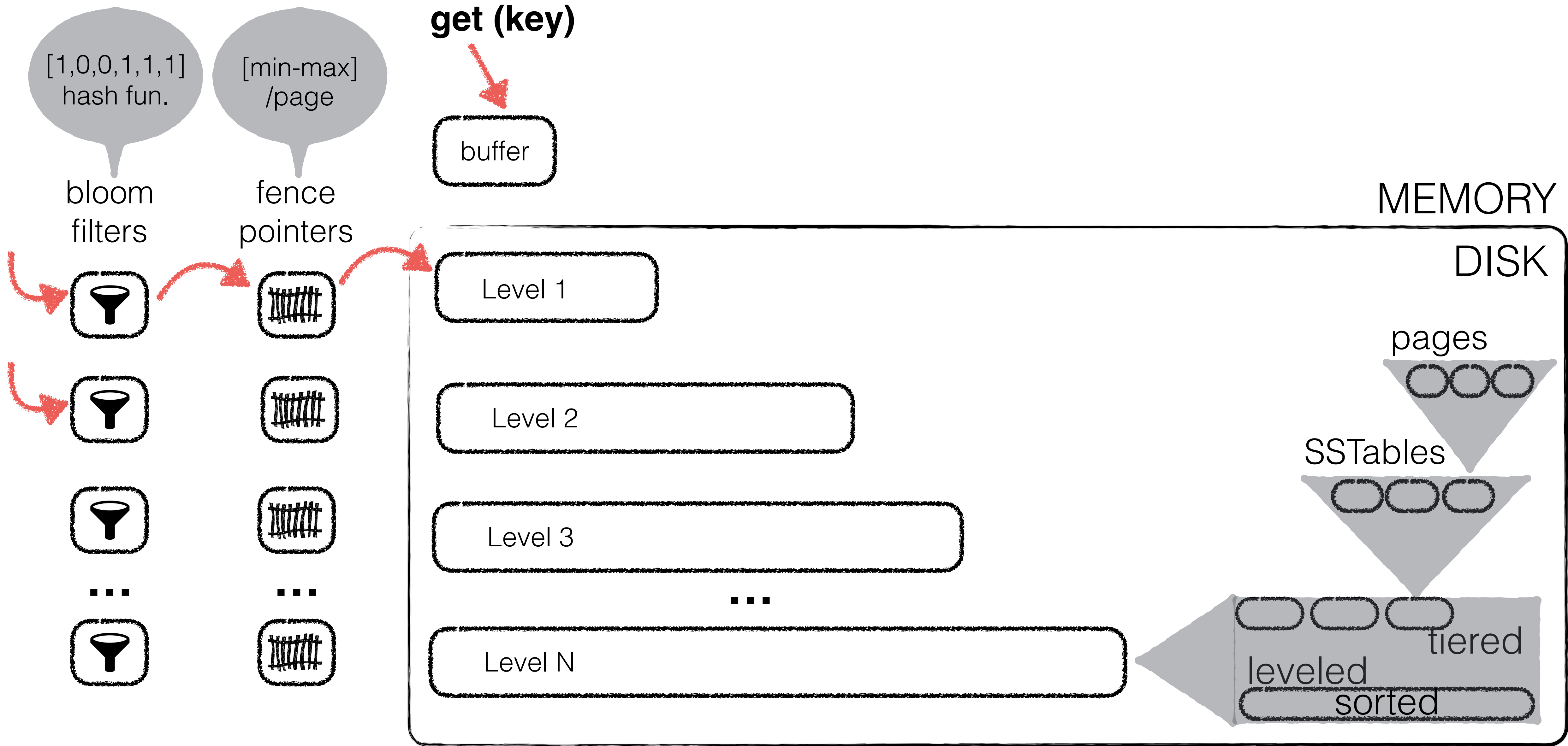
Level 3

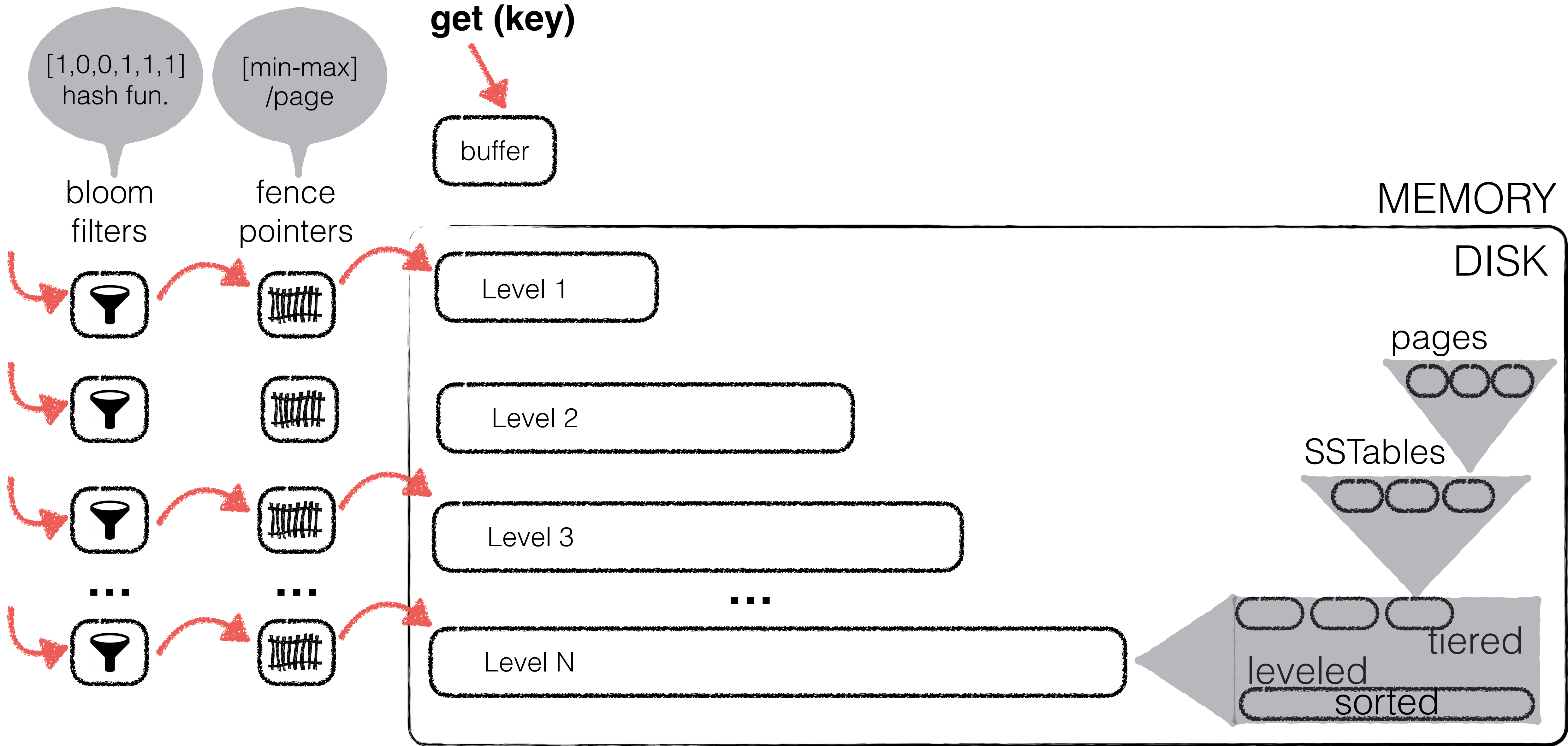
Level N

...



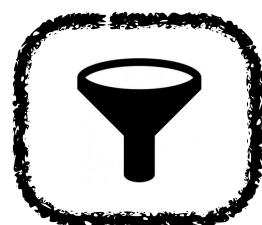






[1,0,0,1,1,1]  
hash fun.

bloom  
filters

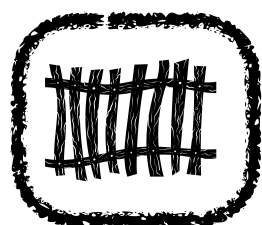
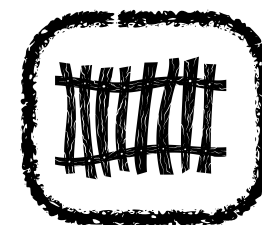
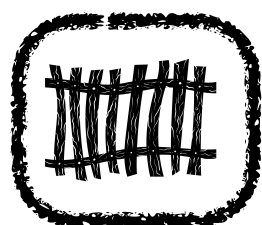


...

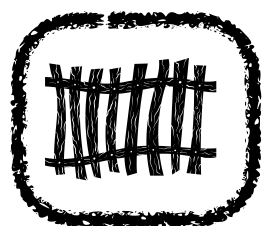


[min-max]  
/page

fence  
pointers



...



buffer

Level 1

Level 2

Level 3

...

Level N

MEMORY  
DISK

pages



SSTables



tiered

leveled

sorted

# Reading for KV background

**Monkey: Optimal Navigable Key-Value Store.** Niv Dayan, Manos Athanassoulis, Stratos Idreos. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 2017

## **Modern B-Tree Techniques**

by Goetz Graefe

Foundations and Trends in Databases, 2011

Sections: 1,2,3,5



CS 265

*Stratos Idreos*

BIG DATA SYSTEMS

NoSQL | Neural Networks | Image AI | LLMs | Data Science