WASHINGTON STATE UNIVERSITY

CPTS 223 Advanced Data Structure C/C++

Data Structure for Artificial Intelligence

Overview

- Generic top-k selection problem and generic algorithms
- Does AI require top-k selection?
- What is new in AI's top-k selection problems?
- Some top-k selection solutions in AI systems
- Other problems in AI improved by better data structures?

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (or k-th largest) number from the input

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (~~or k-th largest~~) number from the input

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (~~or k-th largest~~) number from the input

Given in Chapter 1.1

```
Algorithm 1A
1. Read N numbers into an array A
2. Sort A from smallest to largest one
3. Return the element at k
```

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (~~or k-th largest~~) number from the input

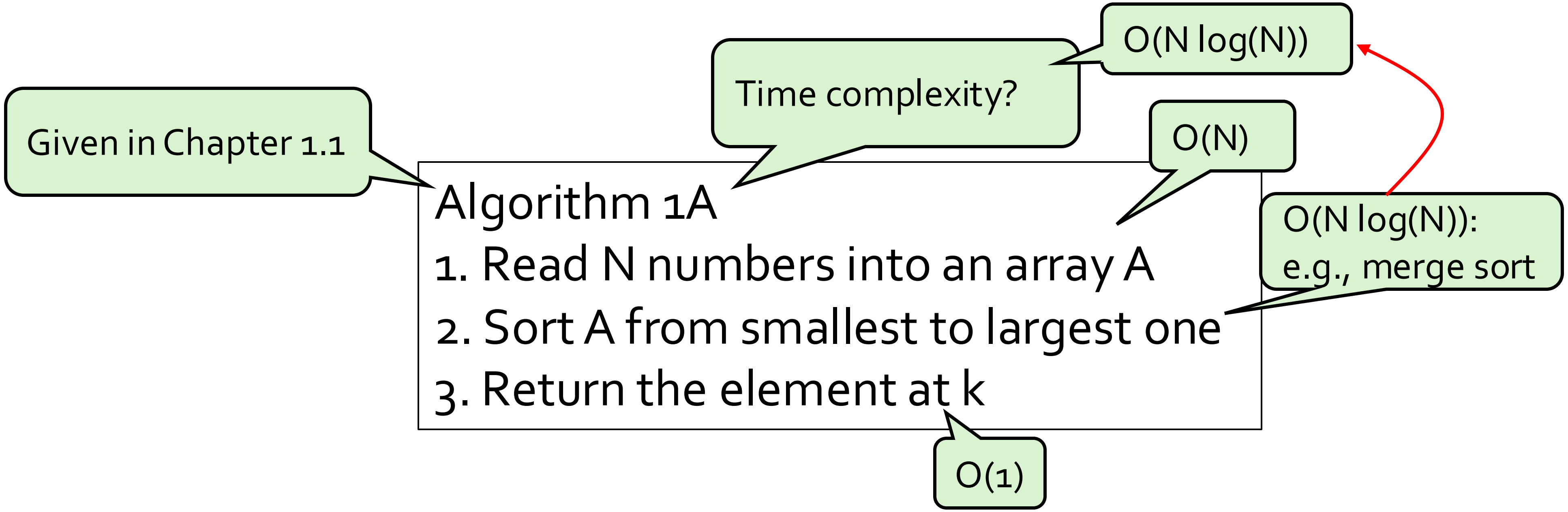
Time complexity?

Given in Chapter 1.1

```
Algorithm 1A
1. Read N numbers into an array A
2. Sort A from smallest to largest one
3. Return the element at k
```

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (~~or k-th largest~~) number from the input



Given in Chapter 1.1

Time complexity?

```
Algorithm 1A
1. Read N numbers into an array A
2. Sort A from smallest to largest one
3. Return the element at k
```

O(N log(N))

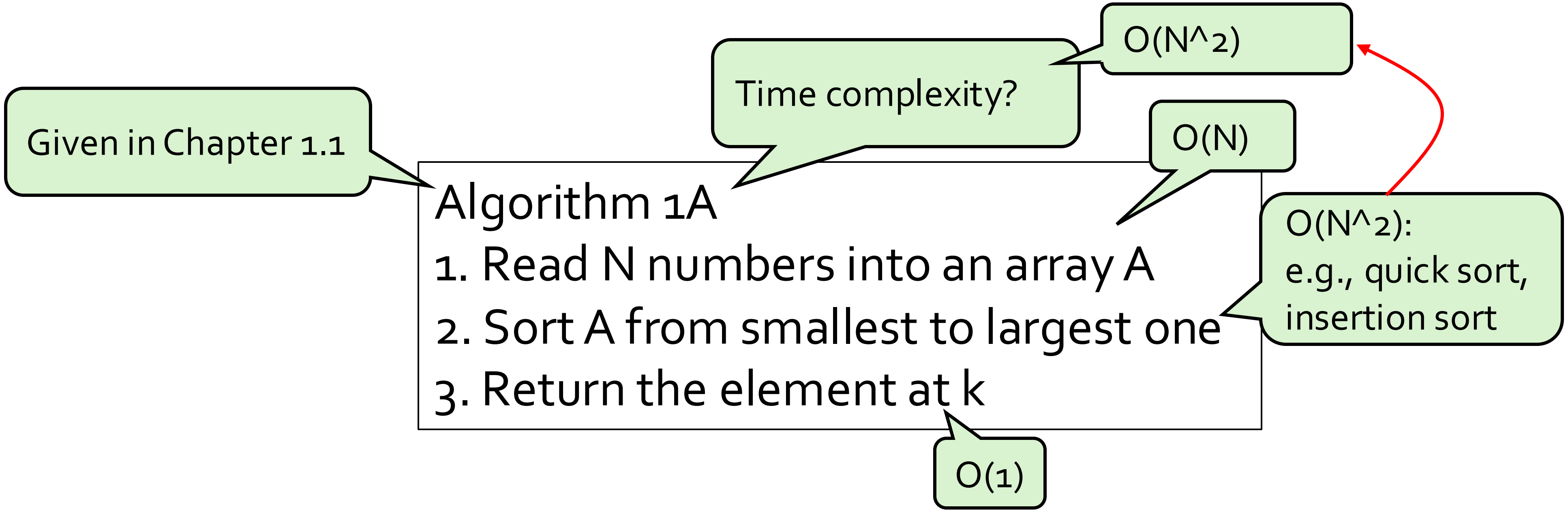
O(N)

O(N log(N)): e.g., merge sort

O(1)

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (~~or k-th largest~~) number from the input



Given in Chapter 1.1

Time complexity?

O(N^2)

O(N)

```
Algorithm 1A
1. Read N numbers into an array A
2. Sort A from smallest to largest one
3. Return the element at k
```

O(N^2):
e.g., quick sort,
insertion sort

O(1)

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (~~or k-th largest~~) number from the input

Given in Chapter 6.4.1

Time complexity?

Algorithm 6A

1. Read N numbers into an array A
2. Construct a min-heap with `buildHeap`
3. Perform k times of `deleteMin` operations
4. Return the last element extracted from the heap

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (~~or k-th largest~~) number from the input

Given in Chapter 6.4.1

Time complexity?

O(N log(N))

```
Algorithm 6A
1. Read N numbers into an array A
2. Construct a min-heap with buildHeap
3. Perform k times of deleteMin operations
4. Return the last element extracted from the heap
```

O(N)

O(N log(N))

O(k)?

O(1)

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (~~or k-th largest~~) number from the input

Given in Chapter 6.4.1

Time complexity?

O(N log(N))

```
Algorithm 6A
1. Read N numbers into an array A
2. Construct a min-heap with buildHeap
3. Perform k times of deleteMin operations
4. Return the last element extracted from the heap
```

O(N)

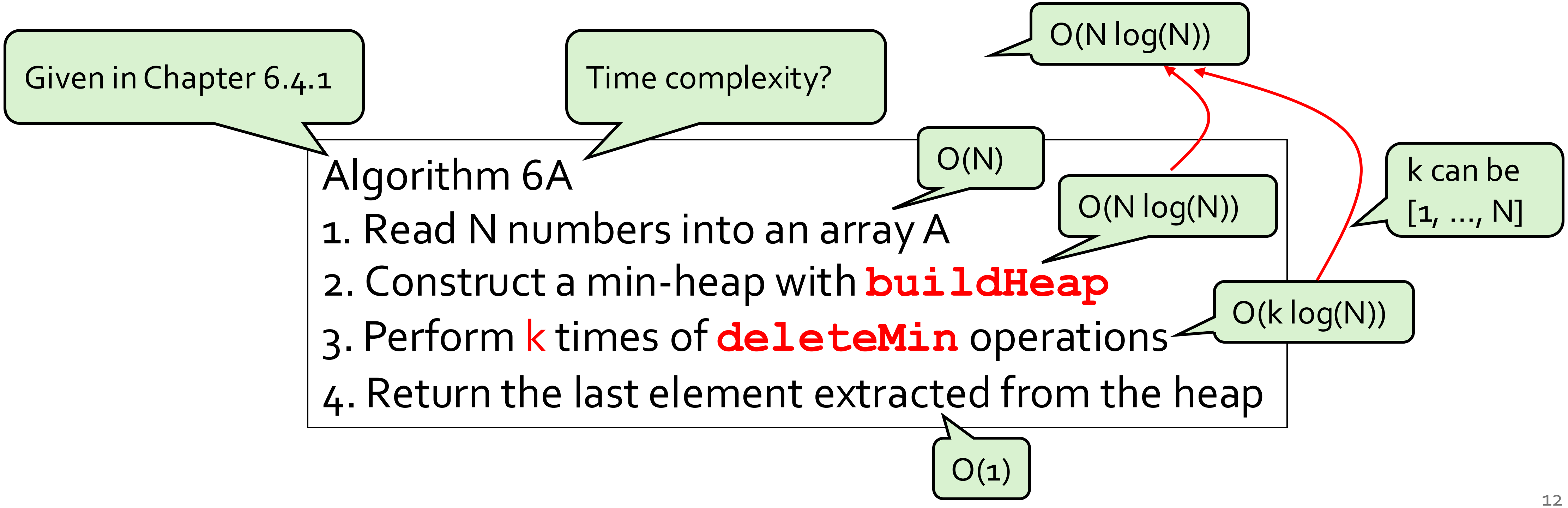
O(N log(N))

O(k log(N))

O(1)

Top-k selection: generic

- Input: a group of N numbers
- Output: the k-th smallest (~~or k-th largest~~) number from the input



Given in Chapter 6.4.1

Time complexity?

O(N log(N))

Algorithm 6A

1. Read N numbers into an array A
2. Construct a min-heap with **buildHeap**
3. Perform k times of **deleteMin** operations
4. Return the last element extracted from the heap

O(N)

O(N log(N))

O(k log(N))

k can be [1, …, N]

O(1)

Data Structure for Artificial Intelligence

Top-k selection in AI?

Self-driving system

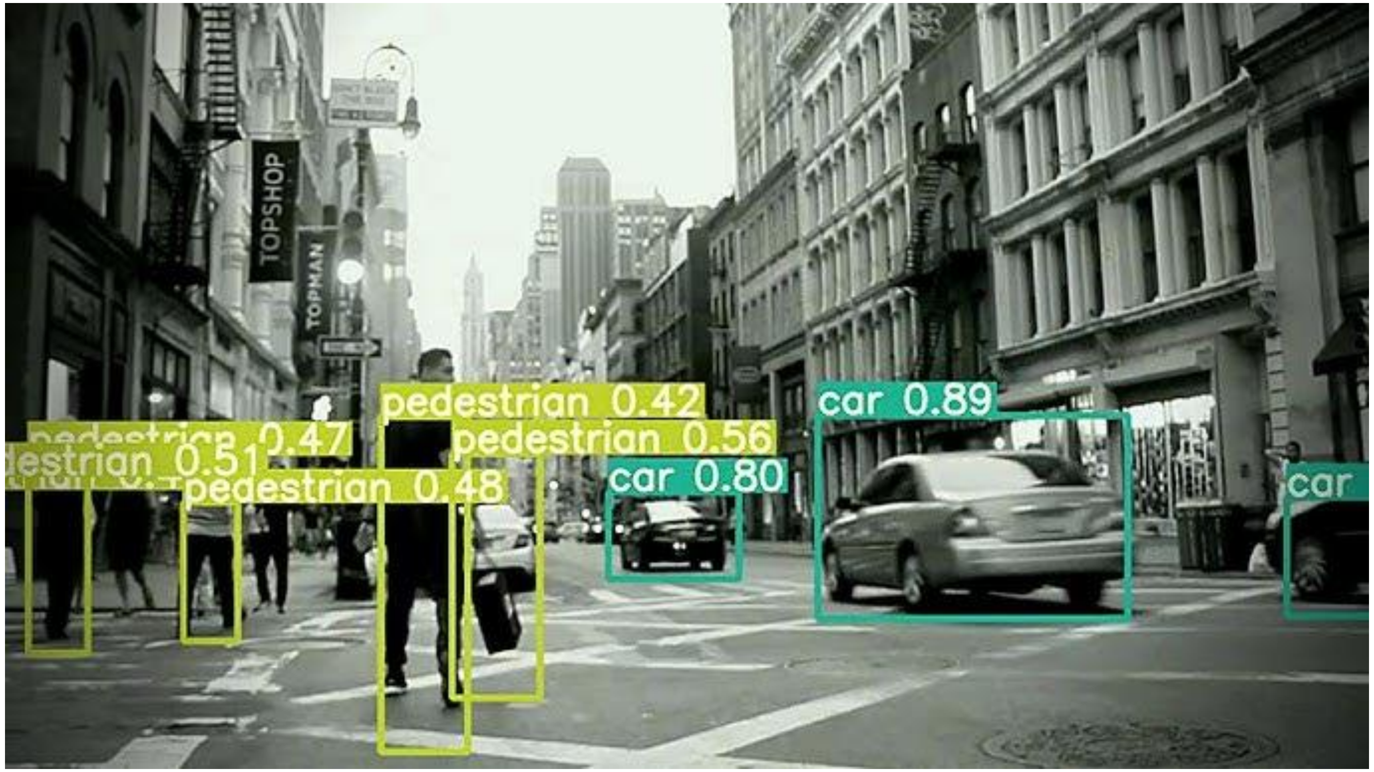


Image credit: https://www.youtube.com/watch?v=fKXztwtXaGo

Top-k selection in AI?

Robotics

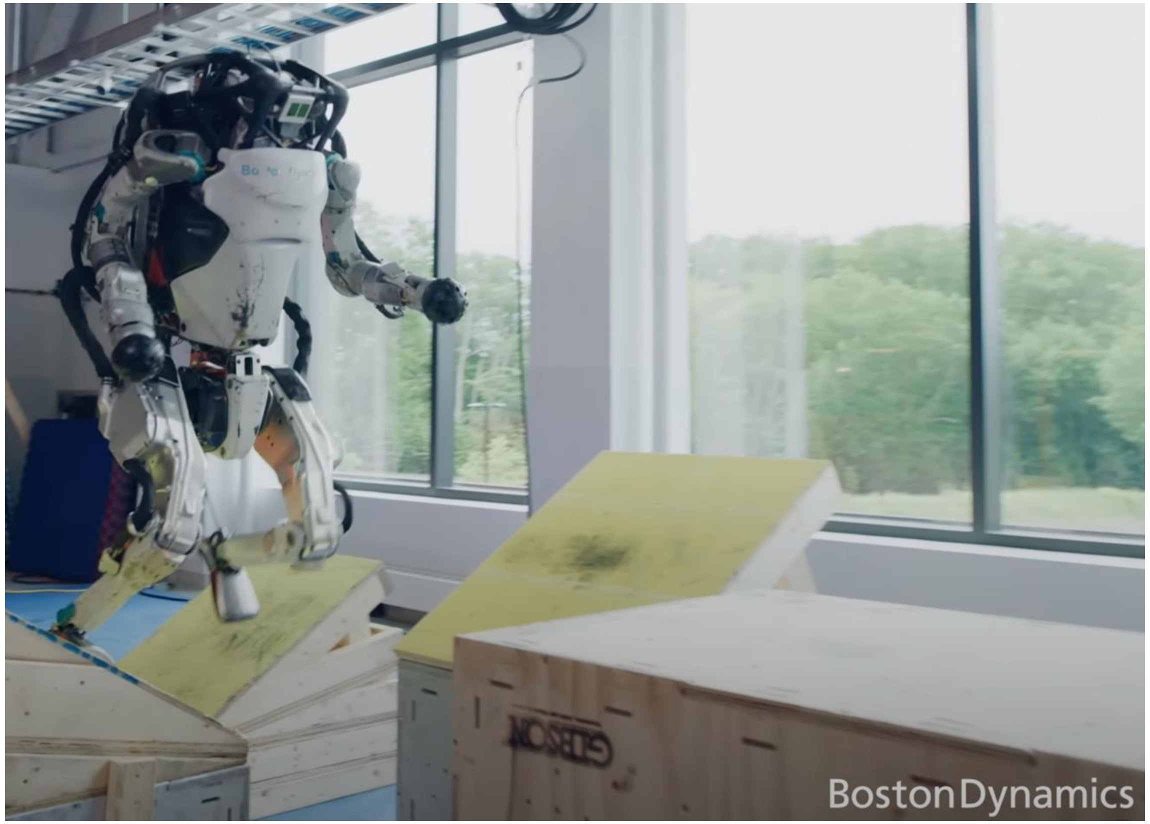
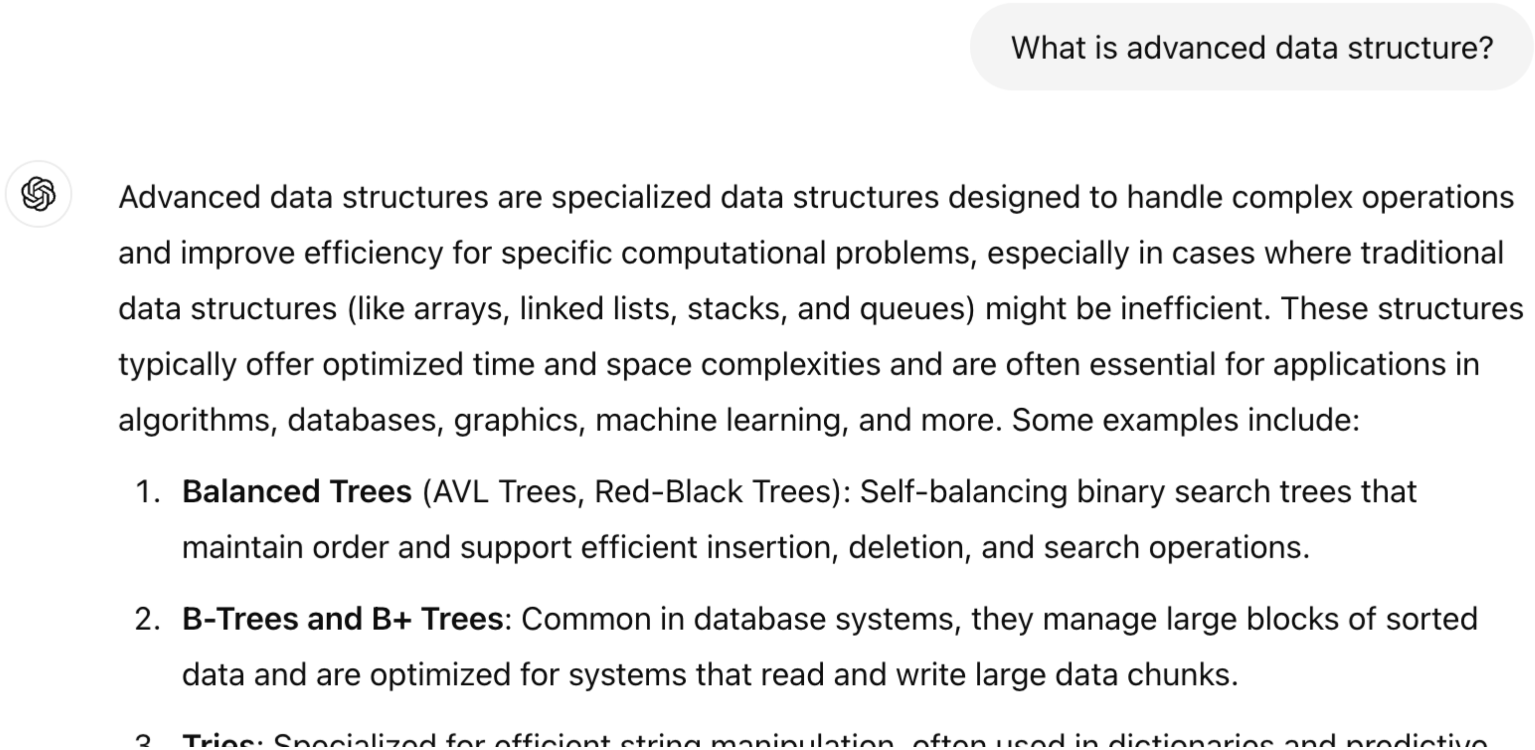


Image credit: https://www.youtube.com/watch?v=tF4DML7FIWk

Top-k selection in AI?

Large language models



What is advanced data structure?

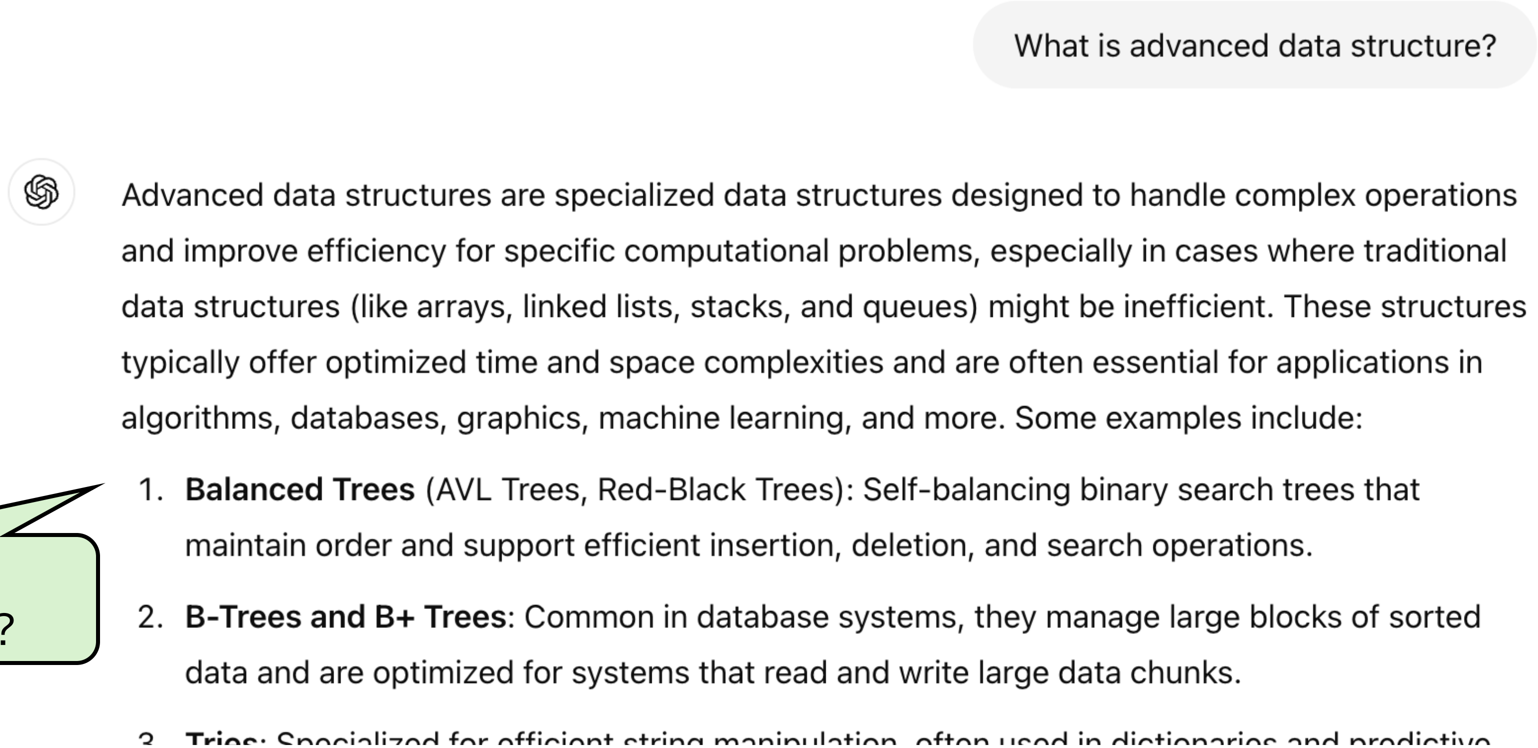
Advanced data structures are specialized data structures designed to handle complex operations and improve efficiency for specific computational problems, especially in cases where traditional data structures (like arrays, linked lists, stacks, and queues) might be inefficient. These structures typically offer optimized time and space complexities and are often essential for applications in algorithms, databases, graphics, machine learning, and more. Some examples include:

1. **Balanced Trees** (AVL Trees, Red-Black Trees): Self-balancing binary search trees that maintain order and support efficient insertion, deletion, and search operations.
2. **B-Trees and B+ Trees**: Common in database systems, they manage large blocks of sorted data and are optimized for systems that read and write large data chunks.
3. **Tries**: Specialized for efficient string manipulation, often used in dictionaries and predictive

Image credit: https://chatgpt.com/

Top-k selection in AI?

Large language models



What is advanced data structure?

Advanced data structures are specialized data structures designed to handle complex operations and improve efficiency for specific computational problems, especially in cases where traditional data structures (like arrays, linked lists, stacks, and queues) might be inefficient. These structures typically offer optimized time and space complexities and are often essential for applications in algorithms, databases, graphics, machine learning, and more. Some examples include:

1. **Balanced Trees** (AVL Trees, Red-Black Trees): Self-balancing binary search trees that maintain order and support efficient insertion, deletion, and search operations.
2. **B-Trees and B+ Trees**: Common in database systems, they manage large blocks of sorted data and are optimized for systems that read and write large data chunks.
3. **Tries**: Specialized for efficient string manipulation, often used in dictionaries and predictive

Does data structure really help modern AI?

Image credit: https://chatgpt.com/

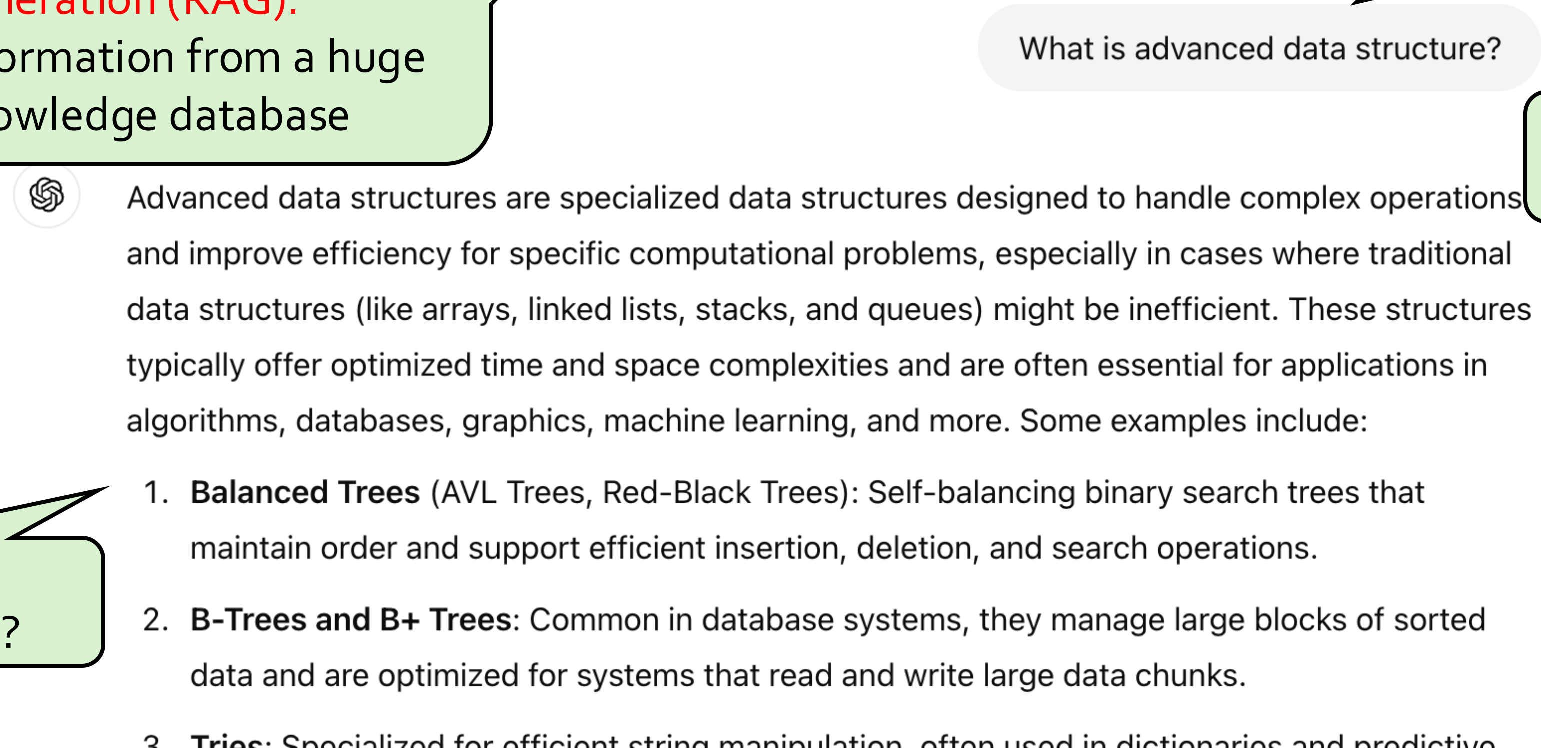
Data Structure for Artificial Intelligence

Top-k selection in AI?

Retrieval-augmented generation (RAG): Information from a huge knowledge database

Large language models

My question: simple



What is advanced data structure?

Advanced data structures are specialized data structures designed to handle complex operations and improve efficiency for specific computational problems, especially in cases where traditional data structures (like arrays, linked lists, stacks, and queues) might be inefficient. These structures typically offer optimized time and space complexities and are often essential for applications in algorithms, databases, graphics, machine learning, and more. Some examples include:

1. **Balanced Trees** (AVL Trees, Red-Black Trees): Self-balancing binary search trees that maintain order and support efficient insertion, deletion, and search operations.
2. **B-Trees and B+ Trees**: Common in database systems, they manage large blocks of sorted data and are optimized for systems that read and write large data chunks.
3. **Tries**: Specialized for efficient string manipulation, often used in dictionaries and predictive

LLM's response:
A lot of information

Does data structure really help modern AI?

Image credit: https://chatgpt.com/

Data Structure for Artificial Intelligence

Top-k selection in AI: LLMs

- Case 1: token generation:
  - Top-k selection for the next token at each step

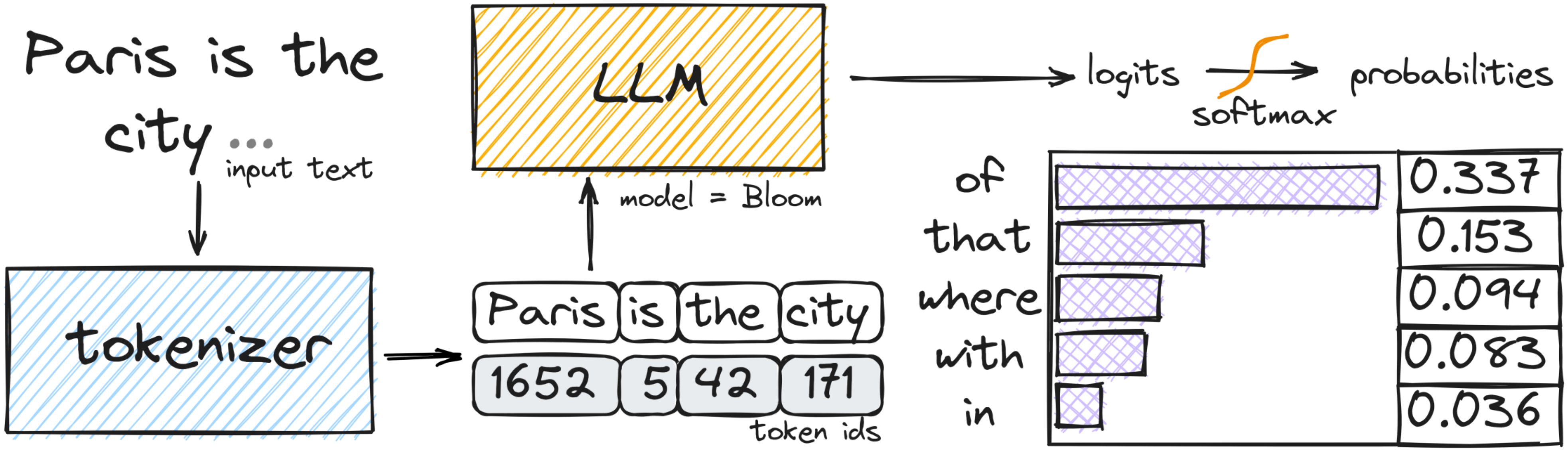
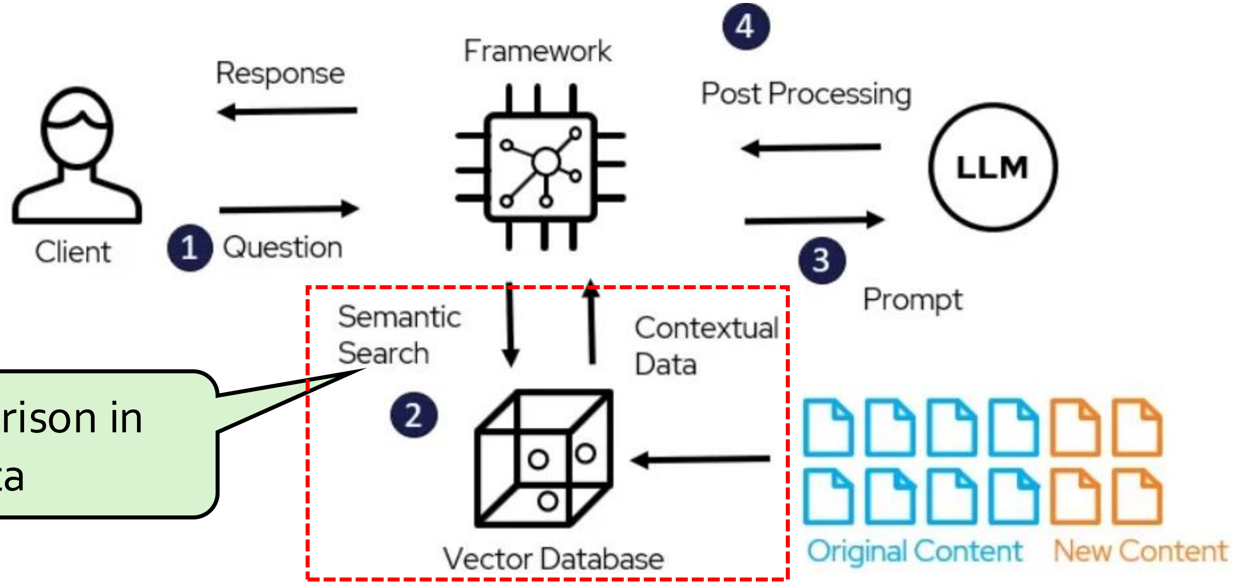


Image credit: https://www.linkedin.com/pulse/how-exactly-llm-generates-text-ivan-reznikov/

Top-k selection in AI: LLMs

- Case 2: retrieval-augmented generation (RAG)
  - Step 1: find the most relevant contextual data from an external database
  - Step 2: generate a final response by combining the original query and the retrieved contextual data
  - → answers are grounded in external knowledge



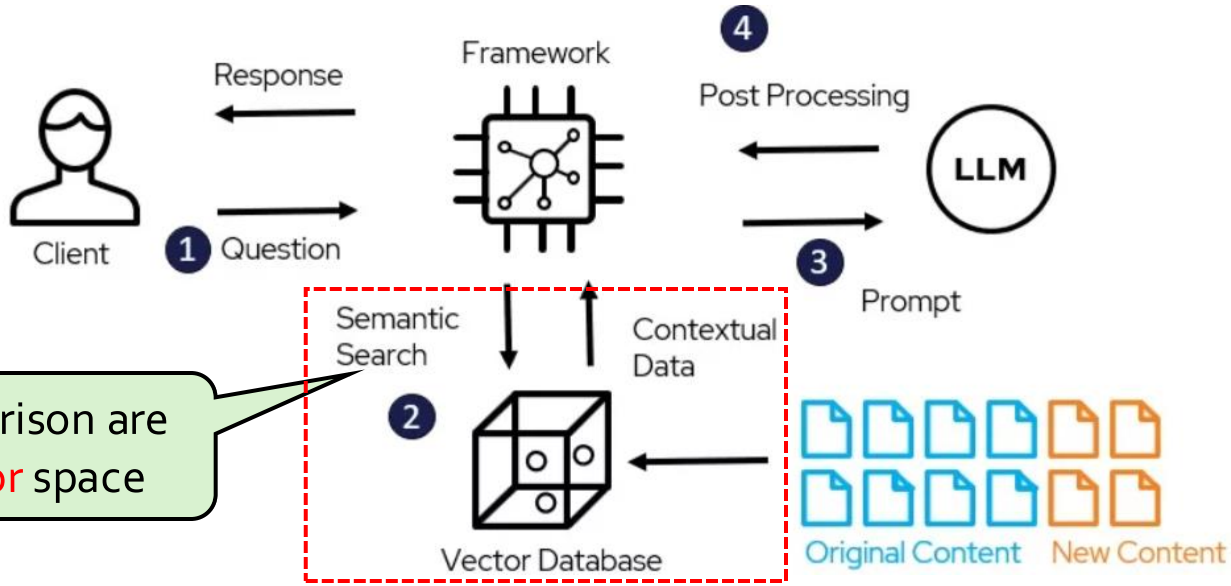
Search and comparison in huge language data

Image credit: https://medium.com/@bijit211987/advanced-rag-for-llms-slms-5bcc6fbba411

Contextual data: format

- Saved as document embeddings in the external database
- → the vector format in $\mathbb{R}^d$

A sentence/document → a vector



Search and comparison are based on the vector space

Image credit: https://medium.com/@bijit211987/advanced-rag-for-llms-slms-5bcc6fbba411

What is new in AI for k-selection?

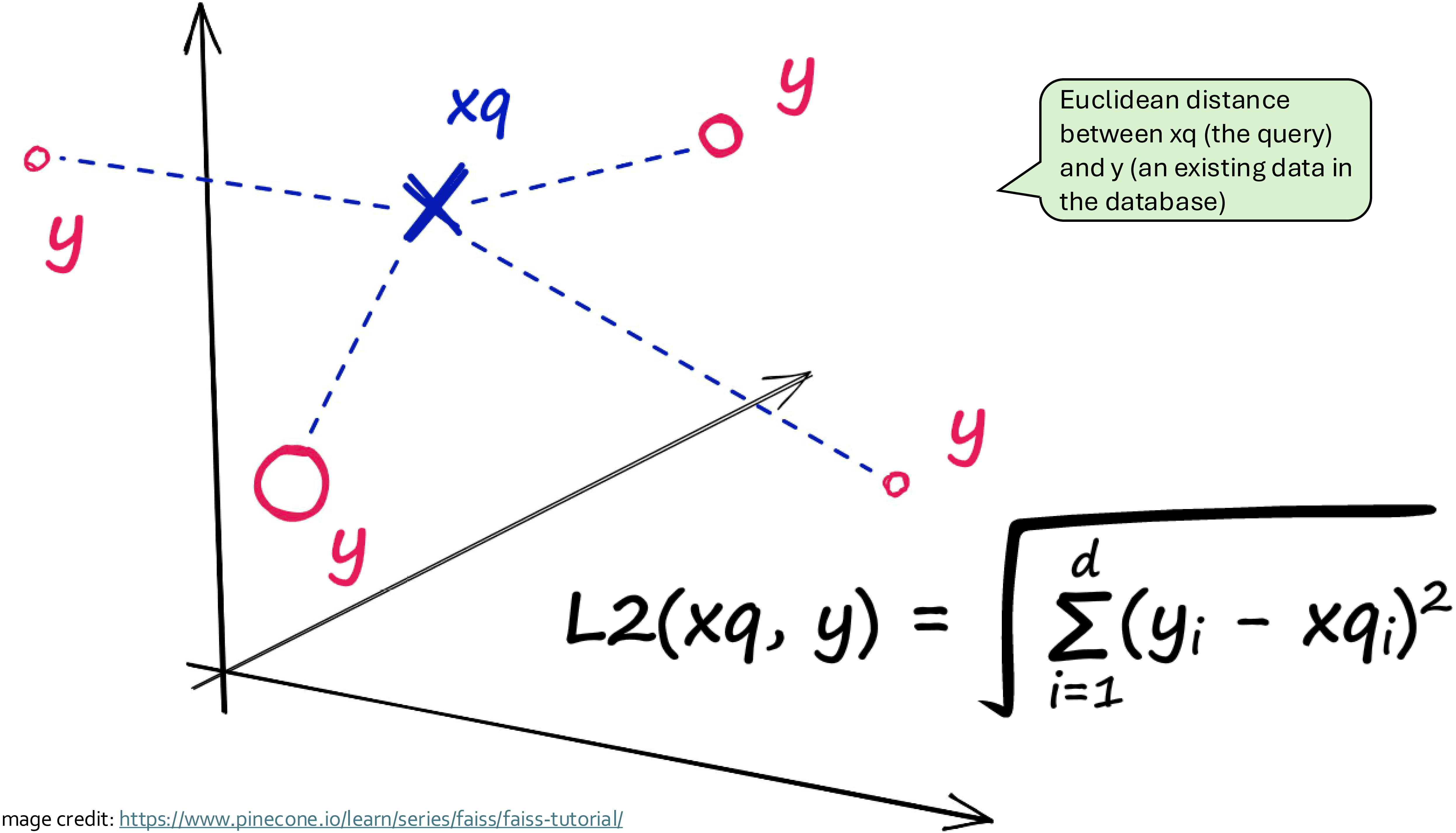
- Generic top-k selection: every element is a real number
  - Scalar (single dimension)
- In LLMs (with RAG)
  - every element is a real-valued vector
    - Vector: multi-dimension
  - The selection criterion:
    - How (semantically) Relevant between the original query text and external database
    - Pairwise measurement, Euclidean distance
      - $R_i := R(A, B_i),\ i \in \{1, 2, \ldots, N\}$
    - Select the top-k most relevant B_i's
    - → top-k selection from $\{R_i\}_{i=1}^{N}$
    - What about we have many A's?

256-1024 dimensions (denoted by d)

Time complexity of $R(A, B_i)$ usually in O(d)

Google's or OpenAI's knowledge retrieval systems: billions of documents

ChatGPT: 2.3 billion visits in January 2024, average 856 queries per second



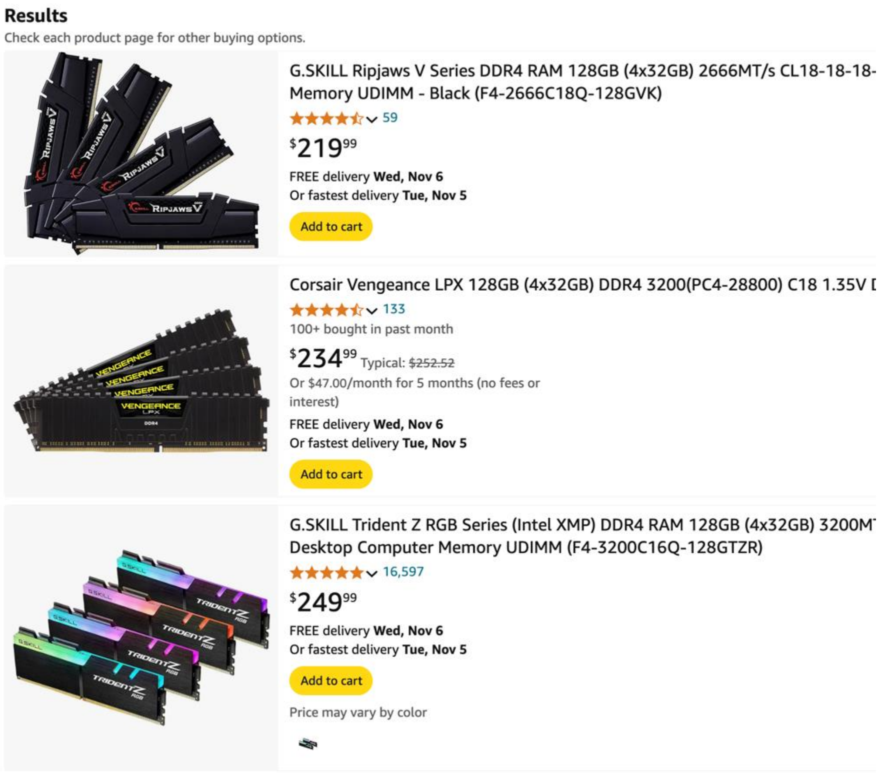
Euclidean distance between xq (the query) and y (an existing data in the database)

$$L2(xq, y) = \sqrt{\sum_{i=1}^{d}(y_i - xq_i)^2}$$

Image credit: https://www.pinecone.io/learn/series/faiss/faiss-tutorial/

What is new in AI for k-selection?

- AI systems are trained and deployed on GPUs
- Challenge 1: small GPU memory
  - CPU memory is large and cheap
  - NVIDIA H100: 80GB, $30K



Results

Check each product page for other buying options.

G.SKILL Ripjaws V Series DDR4 RAM 128GB (4x32GB) 2666MT/s CL18-18-18-
Memory UDIMM - Black (F4-2666C18Q-128GVK)

★★★★☆ 59

$^{\$}219^{99}$

FREE delivery **Wed, Nov 6**

Or fastest delivery **Tue, Nov 5**

Add to cart

Corsair Vengeance LPX 128GB (4x32GB) DDR4 3200(PC4-28800) C18 1.35V I

★★★★☆ 133

100+ bought in past month

$^{\$}234^{99}$ Typical: ~~$252.52~~

Or $47.00/month for 5 months (no fees or interest)

FREE delivery **Wed, Nov 6**

Or fastest delivery **Tue, Nov 5**

Add to cart

G.SKILL Trident Z RGB Series (Intel XMP) DDR4 RAM 128GB (4x32GB) 3200M
Desktop Computer Memory UDIMM (F4-3200C16Q-128GTZR)

★★★★★ 16,597

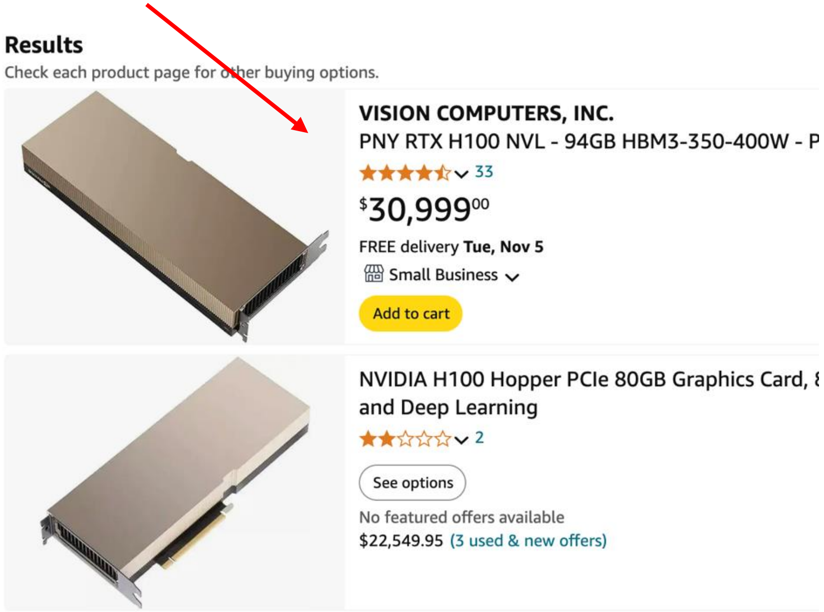
$^{\$}249^{99}$

FREE delivery **Wed, Nov 6**

Or fastest delivery **Tue, Nov 5**

Add to cart

Price may vary by color



Results

Check each product page for other buying options.

VISION COMPUTERS, INC.

PNY RTX H100 NVL - 94GB HBM3-350-400W - P

★★★★☆ 33

$^{\$}30{,}999^{00}$

FREE delivery **Tue, Nov 5**

Small Business

Add to cart

NVIDIA H100 Hopper PCIe 80GB Graphics Card, &
and Deep Learning

★★☆☆☆ 2

See options

No featured offers available

$22,549.95 (3 used & new offers)

What is new in AI for k-selection?

- AI systems are trained and deployed on GPUs
- Challenge 1: small GPU memory
  - CPU memory is large and cheap
  - NVIDIA H100: 80GB
- Challenge 2: slow data transmission between CPU and GPU
  - Within CPU (e.g., RAM)
    - 100+ GB/s
  - Within GPU
    - 100 to 1000 GB/s
  - Between CPU and GPU
    - PCIe, e.g., 32GB/s PCIe 5.0

Data loading from a standard computer architecture:
Hard drive → CPU memory → GPU memory
→ return results to CPU memory

What is new in AI for k-selection?

- AI systems are trained and deployed on GPUs
- Fact : GPU has a very special hierarchy
  - Memory hierarchy
    - Global memory (GB, slow) → shared memory (KB, fast) → registers (very fast)
  - Compute hierarchy
    - Grid → thread block → warp → thread

A thread block has a shared memory: All warps/threads in this block can access to the shared memory

A warp includes 32 threads

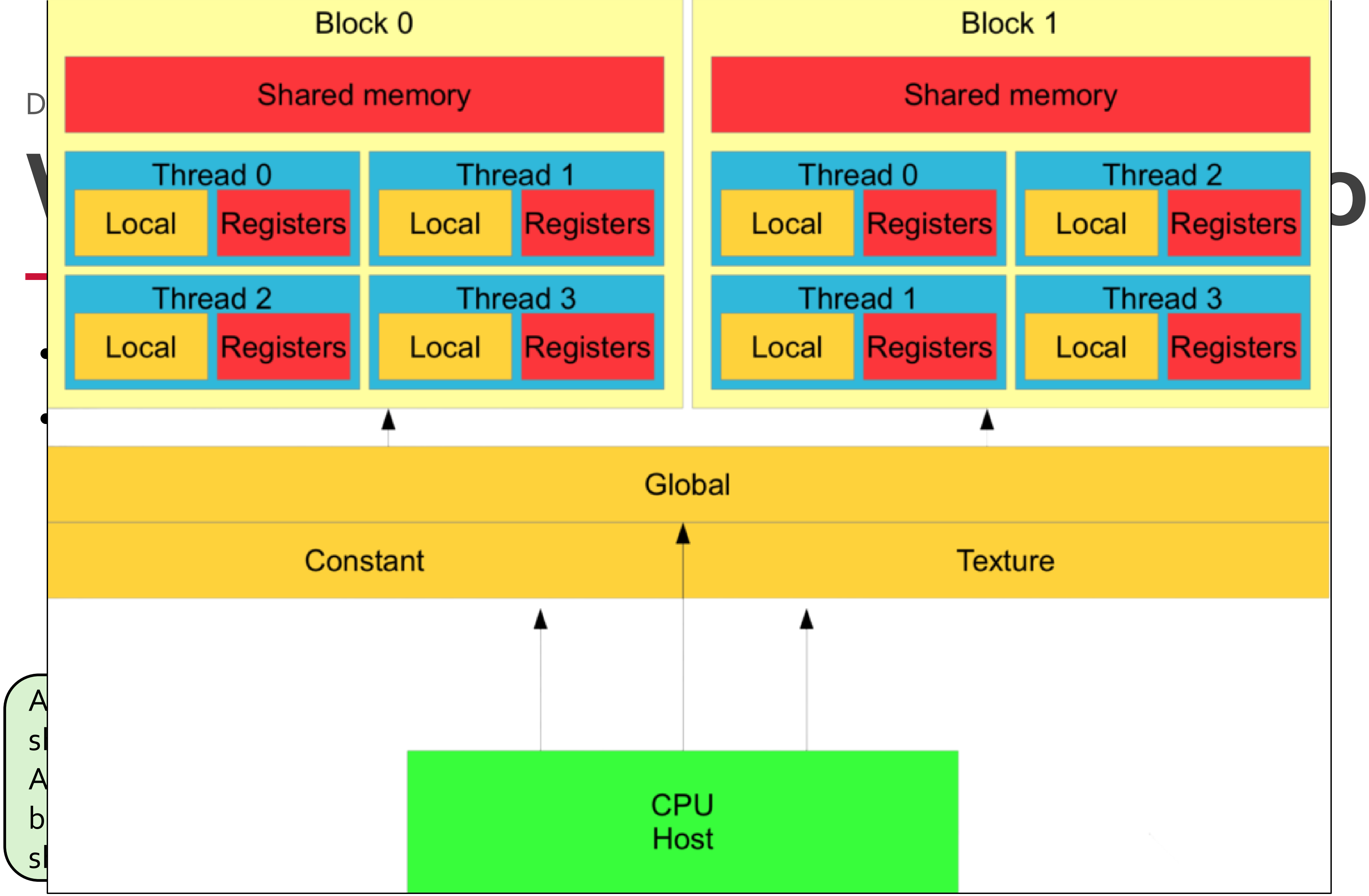


image credit: http://thebeardsage.com/cuda-memory-hierarchy/

Top-k selection in AI: solutions

- k-NN search in RAG:
  - FAISS [1]: a library for efficient similarity search and clustering of dense vectors
  - Original research paper [2]: Billion-scale similarity search with GPUs
  - The design is a combination of many fields:
    - Computer architecture
    - Artificial intelligence models
    - Computation
    - Optimization
    - Statistical approximation
    - Data structure
    - …

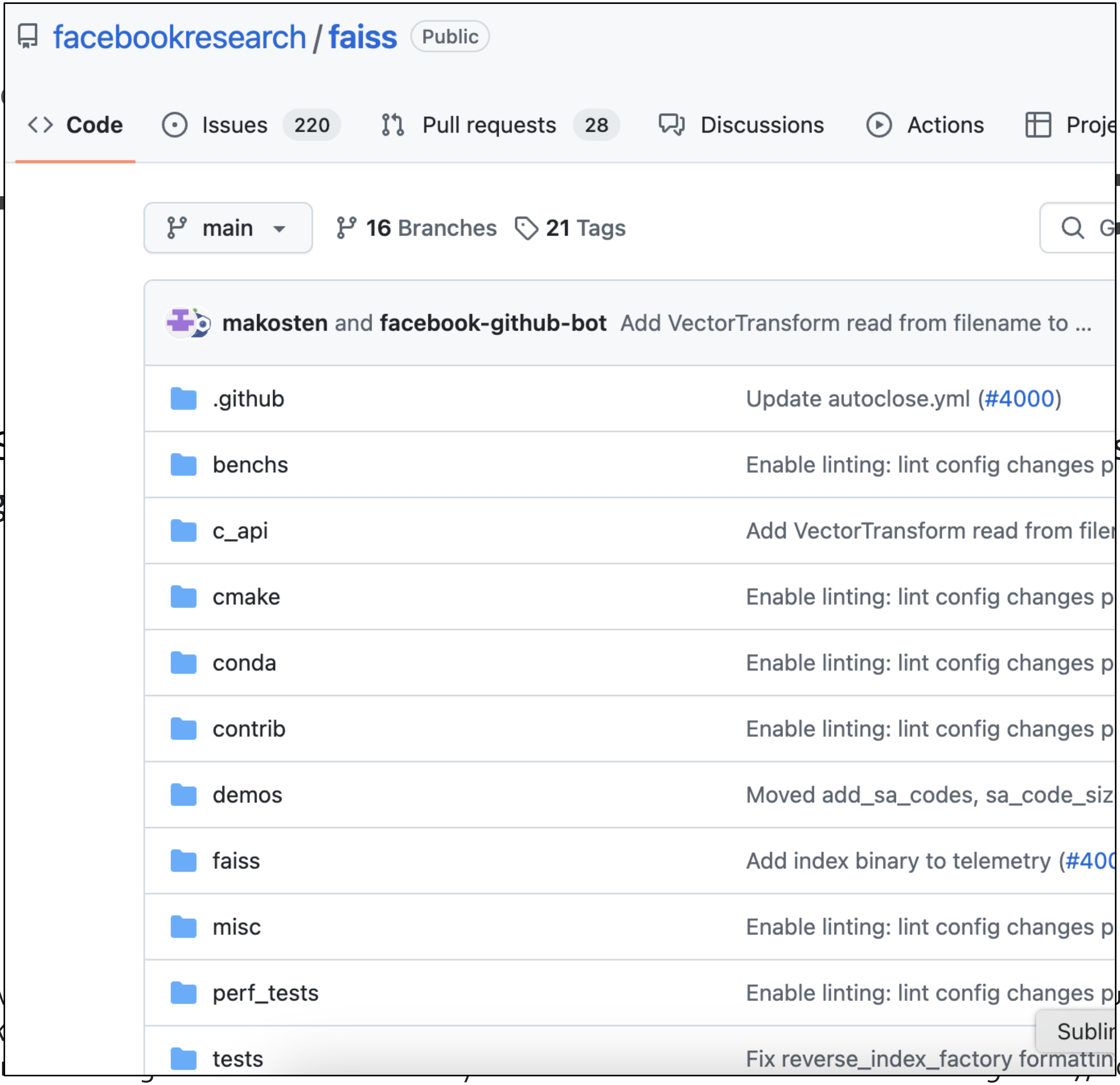
[1] Douze, Matthijs, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. "The faiss library." *arXiv preprint arXiv:2401.08281* (2024). https://github.com/facebookresearch/faiss

[2] Johnson, Jeff, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with GPUs." *IEEE Transactions on Big Data* 7, no. 3 (2019): 535-547.

Data Structur

Top- ions

- k-NN s
  - FAIS se vectors
  - Orig
  - The
    -
    -
    -
    -
    -
    -
    -



facebookresearch / faiss Public

makosten and facebook-github-bot Add VectorTransform read from filename to ...

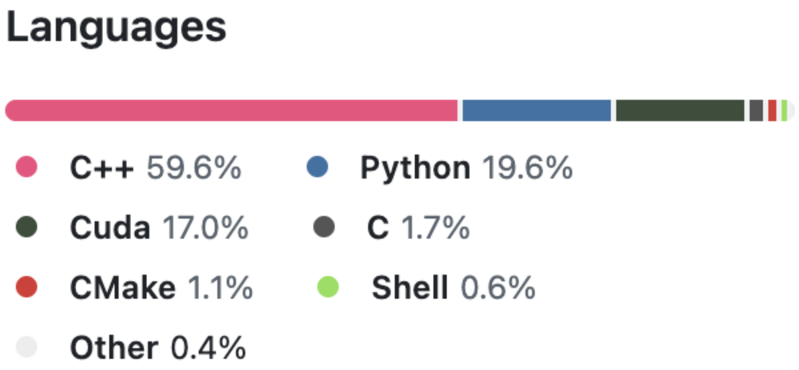
.github	Update autoclose.yml (#4000)
benchs	Enable linting: lint config changes p
c_api	Add VectorTransform read from filer
cmake	Enable linting: lint config changes p
conda	Enable linting: lint config changes p
contrib	Enable linting: lint config changes p
demos	Moved add_sa_codes, sa_code_siz
faiss	Add index binary to telemetry (#400
misc	Enable linting: lint config changes p
perf_tests	Enable linting: lint config changes p
tests	Fix reverse_index_factory formattin

[1] Douze, Matthijs, Alexandr Guzhv ucas Hosseini, and Hervé Jégou. "The faiss library." *arX*

[2] Johnson, Jeff, Matthijs Douze, a o. 3 (2019): 535-547.

Top-k selection in AI: solutions

- k-NN search in RAG:
  - FAISS [1]: a library for efficient similarity search and clustering of dense vectors
  - Original research paper [2]: Billion-scale similarity search with GPUs
  - The design is a combination of many fields:
    - Computer architecture
    - Artificial intelligence models
    - Computation
    - Optimization
    - Statistical approximation
    - Data structure
    - …



Languages

● C++ 59.6%
● Python 19.6%
● Cuda 17.0%
● C 1.7%
● CMake 1.1%
● Shell 0.6%
● Other 0.4%

[1] Douze, Matthijs, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. "The faiss library." *arXiv preprint arXiv:2401.08281* (2024). https://github.com/facebookresearch/faiss

[2] Johnson, Jeff, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with GPUs." *IEEE Transactions on Big Data* 7, no. 3 (2019): 535-547.

Top-k selection in AI: solutions

- k-NN
  - FA … vectors
  - Or

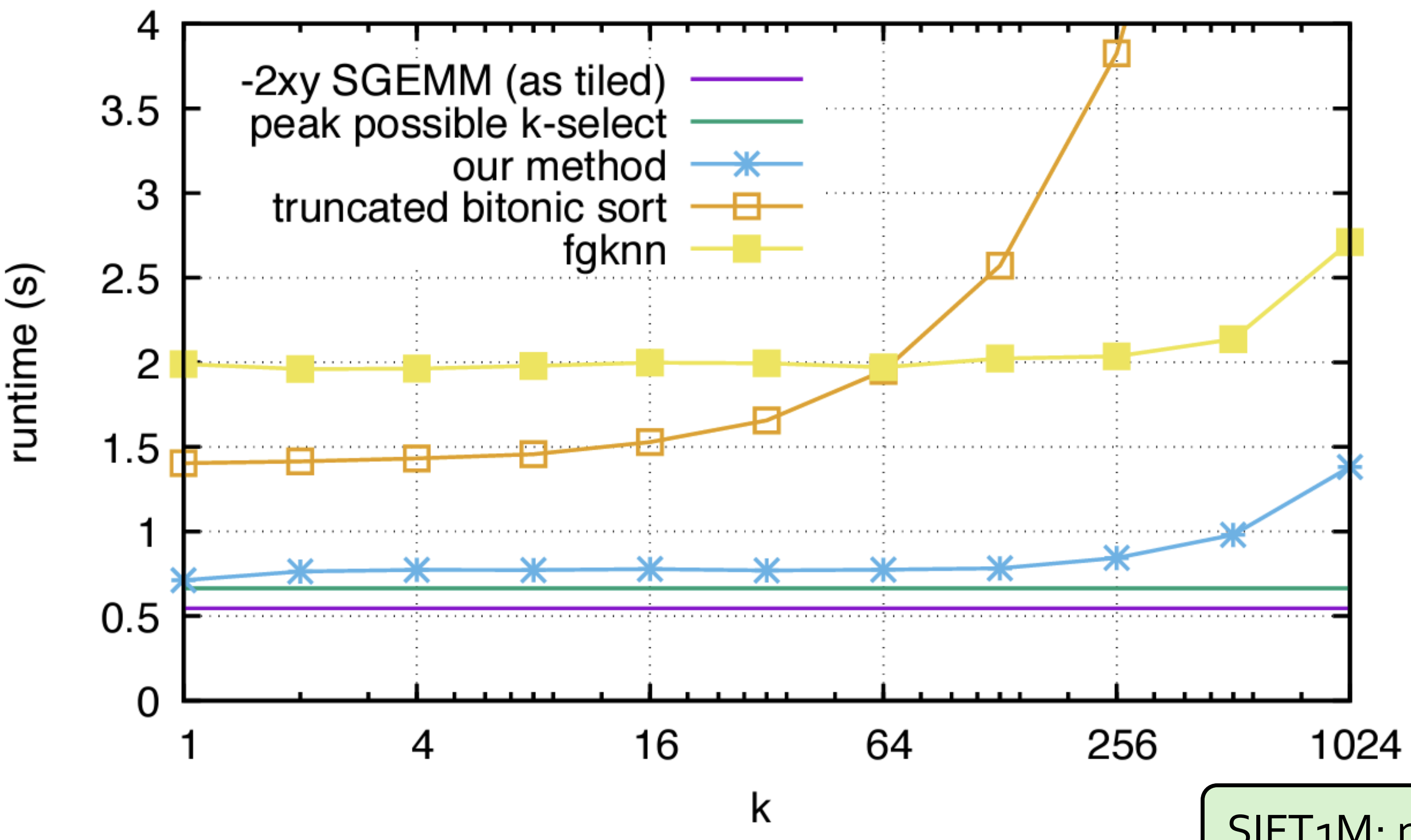


Figure 4: Exact search k-NN time for the SIFT1M dataset with varying k on 1 Titan X GPU.

Process 50000 queries per second

0.02ms per query on a Titan X GPU 12 GB, around $200

SIFT1M: n_q = 10,000 queries

[1] Douze, Matthijs, Alexandr Guz … Hosseini, and Hervé Jégou. "The faiss library." *a*…

[2] Johnson, Jeff, Matthijs Douze, … (2019): 535-547.