Big Data Computing

Master's of Science in Computer Science

Sapienza University of Rome

2026/2027

Gabriele Tolomei

tolomei@di.uniroma1.it

Module 3: Scaling Classical Tasks

Lecture 3.3 (Part I)

Finding Similar Items: From Documents to Signatures

Philosophy of Similarity Search

> *To find similar needles in a haystack of ten billion straws, first make every straw small, then make sure you never compare straws that have nothing in common.*

*Today we do the first half: shrink each document into a short fingerprint that still remembers **how similar** it is to every other document. Next time we do the second half: avoid comparing most pairs.*

Where We Are: One Crawl, Four Lectures

Lecture	Question about the crawl	The answer is	Exact?
3.1 Search	Which pages match this query?	A top-k list	Exact
3.2 PageRank	Which pages are authoritative?	A score per page	Exact (iterative)
3.3 Similarity (today: Part I)	Which pages are near-duplicates?	A list of pairs	**Approximate**
3.4 AQP	What do aggregates of the crawl look like?	A number	**Approximate**, with guarantee

Lecture 3.3 is where the module turns: still a list of items as the answer, but exact computation is no longer affordable.

Lecture Roadmap

Part I builds the first half of near-duplicate detection, in **5 sections**:

1. **The Problem** — near-duplicates, and why all pairs is hopeless
2. **Documents as Sets** — shingles and Jaccard similarity
3. **MinHash** — a random fingerprint whose collisions measure similarity
4. **Signatures** — many fingerprints, a worked example, how accurate, how cheap
5. **Synthesis and Bridge** — what remains for Part II

> *Extended material (full proofs, hash families in practice, Spark code, exam problems) is in the **Appendix** (A.1–A.5).*

§ 1 — The Problem

Near-duplicates in the crawl

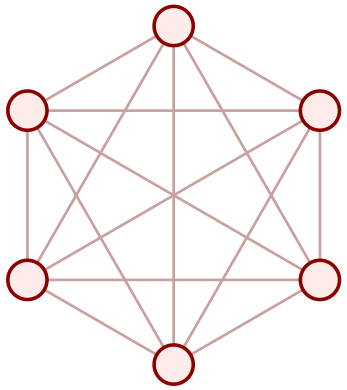
📎 *Extended material → Appendix A.1*

From Keywords to Items

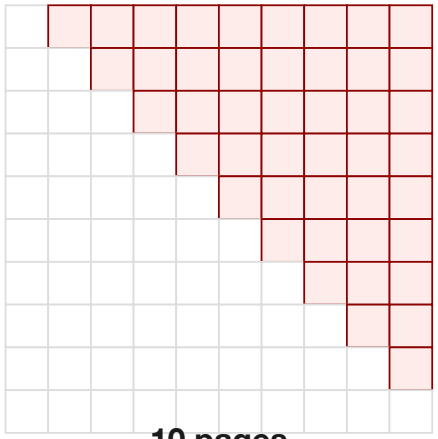
- In Lecture 3.1 the query was a few words, and an inverted index answered it exactly.
- Now the "query" is **a whole page**: *which other pages are near-duplicates of this one?*
- Near-duplicates are everywhere: mirrors, syndicated articles, pages that differ only in a banner or a date.
- They waste crawl budget, clutter search results, and split link votes (Lecture 3.2).

The inverted index cannot help: a whole page shares words with almost every other page.

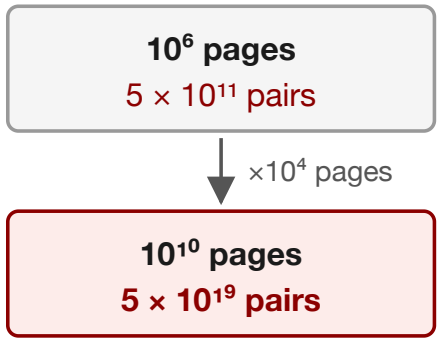
All Pairs: How Many?



6 pages
15 pairs



10 pages
45 pairs



pairs grow as n^2
($\times 10^8$ more pairs)

- 10^{10} pages give about 5×10^{19} pairs. At 10^{12} comparisons per second: **about 1.6 years**.
- And each comparison means reading **both** pages.

Looking at every pair is not an option, at any price.

What "Similar" Means

We need two choices:

- a **similarity** $s(A, B) \in [0, 1]$, with 1 meaning identical;
- a **threshold** t, for example $t = 0.8$.

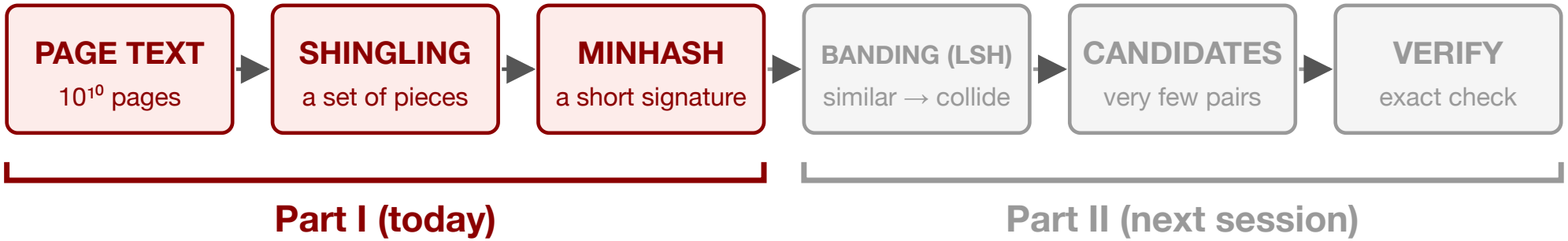
Goal: find the pairs with $s(A, B) \geq t$ **without looking at all pairs**.

- Today's similarity: **Jaccard**, on documents seen as *sets* (§2).
- We accept a small chance of missing a pair. Part II measures it exactly.

Approximation is part of the problem statement from the start.

The Plan: A Pipeline That Shrinks the Problem

Each box makes the data smaller or the pairs fewer



- Each step makes the data **smaller** or the pairs **fewer**.
- The last step checks the few surviving pairs exactly.

Today: the first two boxes.

✅ Checkpoint 1 — The Problem

Test your understanding:

1. How is the query of this lecture different from the query of Lecture 3.1?
2. How many pairs does a collection of 10^6 pages have? And of 10^{10}?
3. Why is "compare every pair" hopeless even with $1{,}000$ machines?
4. What two things do we fix to define the goal precisely?

Consolidate your answers before proceeding.

§ 2 — Documents as Sets

Shingles and Jaccard similarity

📎 *Extended material → Appendix A.2*

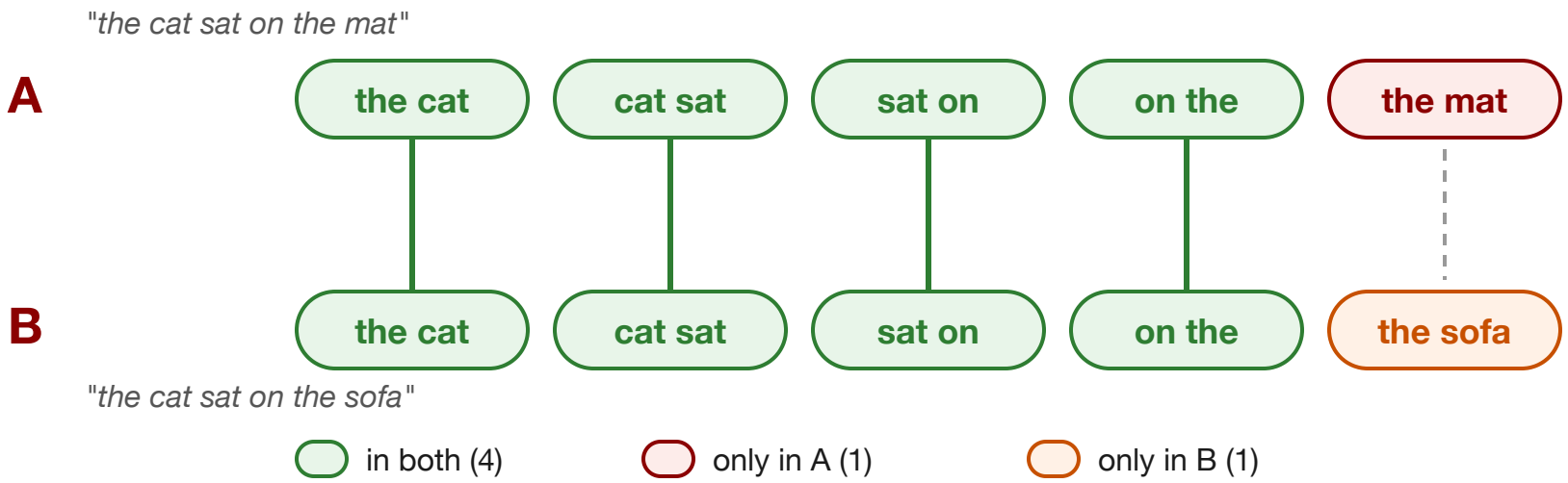
A Document as a Set of Pieces



- Cut the text into overlapping pieces of k consecutive words: the k-**shingles**.
- Nearly identical documents share **most** of their shingles.

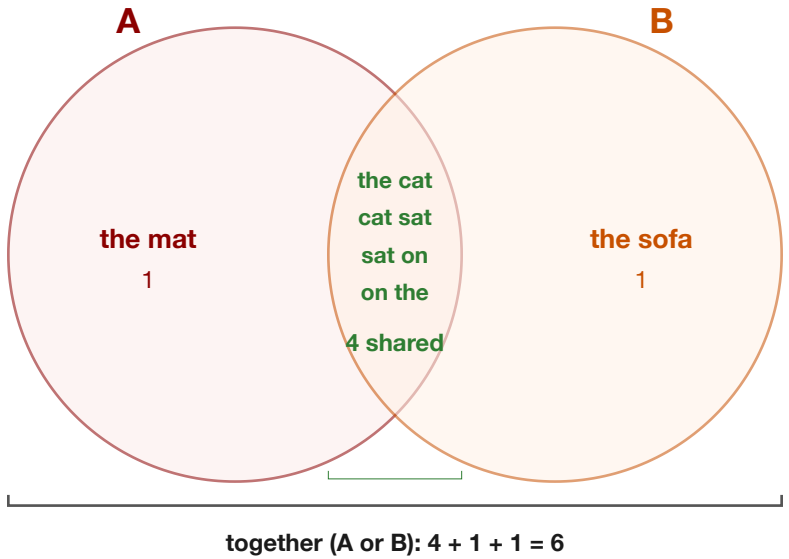
A document becomes a set. Comparing documents becomes comparing sets.

Our Running Example: Shingles with $k = 2$ Words



Two sentences, one word of difference. We keep using exactly these two documents.

Counting What Is Shared



Shared over together: this is the similarity we want.

Jaccard Similarity

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \qquad \text{our example: } J = \frac{4}{6} = \frac{2}{3}$$

- $J = 1$: identical sets. $J = 0$: nothing in common.
- Reads as: *of all the shingles either document has, what fraction do both have?*

From now on, "similar" means "large Jaccard similarity".

The Characteristic Matrix

One **column** per document, one **row** per possible shingle. A 1 means "this document contains this shingle".

shingle	A	B
the cat	1	1
cat sat	1	1
sat on	1	1
on the	1	1
the mat	1	0
the sofa	0	1

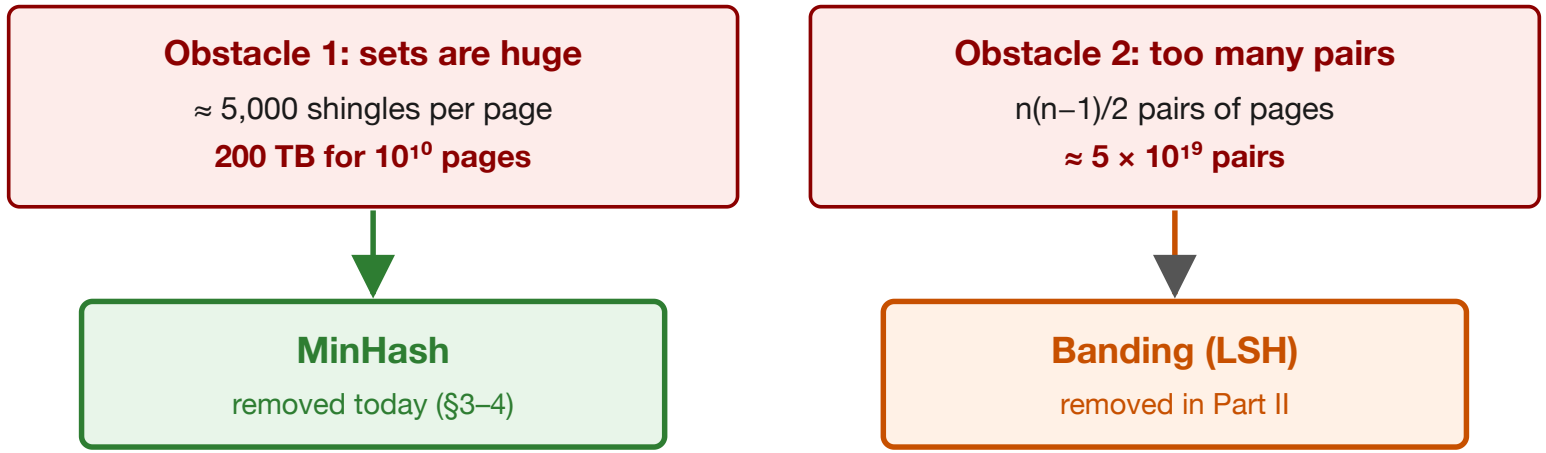
(1, 1): in both
counted in A ∩ B and A ∪ B

(1, 0) or (0, 1): in one
counted in A ∪ B only

(0, 0) rows are ignored

J = 4 / 6

Two Obstacles at Web Scale



Obstacle 1 is the size of each set. Obstacle 2 is the number of pairs.

Choosing the Shingle Length k

- **Too short** (k = 1 or 2 characters): almost every shingle appears in almost every page. All pages look alike.
- **Too long**: tiny edits destroy many shingles. Similar pages look different.
- **Rule of thumb:** $k \approx 5$ characters for short texts, $k \approx 9$–10 for long documents.
- Rare shingles should be rare: the chance that two unrelated pages share one must stay small.

Why these numbers: Appendix A.2.

✅ Checkpoint 2 — Documents as Sets

Test your understanding:

1. List the 2-word shingles of "to be or not to be".
2. Compute J for $A = \{1, 2, 3, 4\}$ and $B = \{3, 4, 5\}$.
3. In the characteristic matrix, which rows are ignored when computing J, and why?
4. What happens to similarity detection if k is chosen far too small?

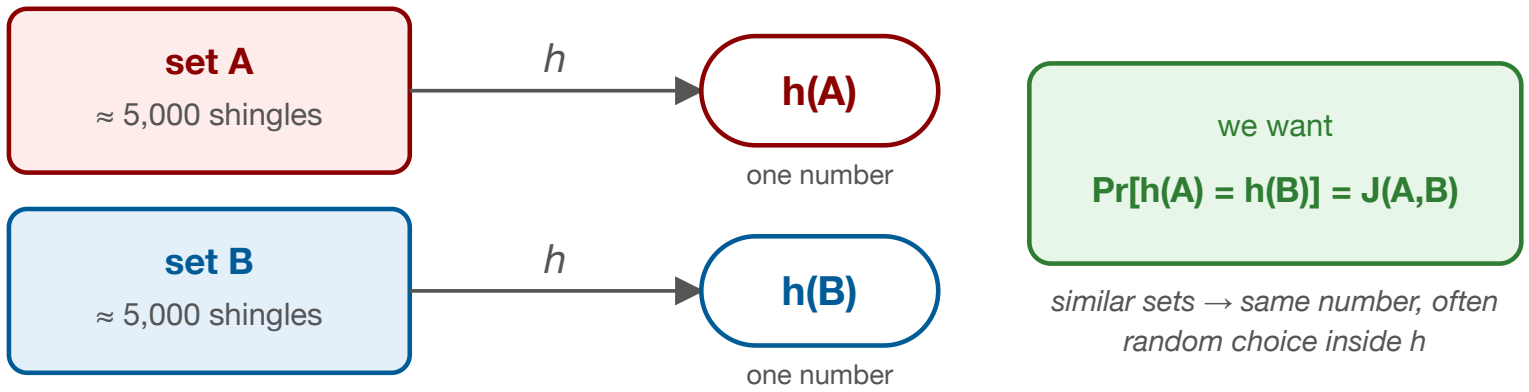
Consolidate your answers before proceeding.

§ 3 — MinHash

A random fingerprint whose collisions measure similarity

📎 *Extended material → Appendix A.3*

Goal: A Short Fingerprint for a Set



Surprisingly, a very simple random choice does exactly this.

Step 1: Put the Rows in a Random Order

original order

	A	B
the cat	1	1
cat sat	1	1
sat on	1	1
on the	1	1
the mat	1	0
the sofa	0	1

random shuffle

same data, new row order

shuffled order π_1

	A	B
the mat	**1**	0
the cat	1	**1**
cat sat	1	1
sat on	1	1
on the	1	1
the sofa	0	1

Step 2: The First Row With a 1

$h_\pi(S)$ = **the first row, in the order π, in which S has a 1.**

	A	B
the mat	**1**	0
the cat	1	**1**
cat sat	1	1
sat on	1	1
on the	1	1
the sofa	0	1

h(A) = the mat

h(B) = the cat

different → no collision

scan down from the top: first 1 in each column

Two More Random Orders

π_1	A	B
the mat	**1**	0
the cat	1	**1**
cat sat	1	1
sat on	1	1
on the	1	1
the sofa	0	1

no collision

π_2	A	B
sat on	**1**	**1**
the sofa	0	1
the mat	1	0
the cat	1	1
cat sat	1	1
on the	1	1

collision

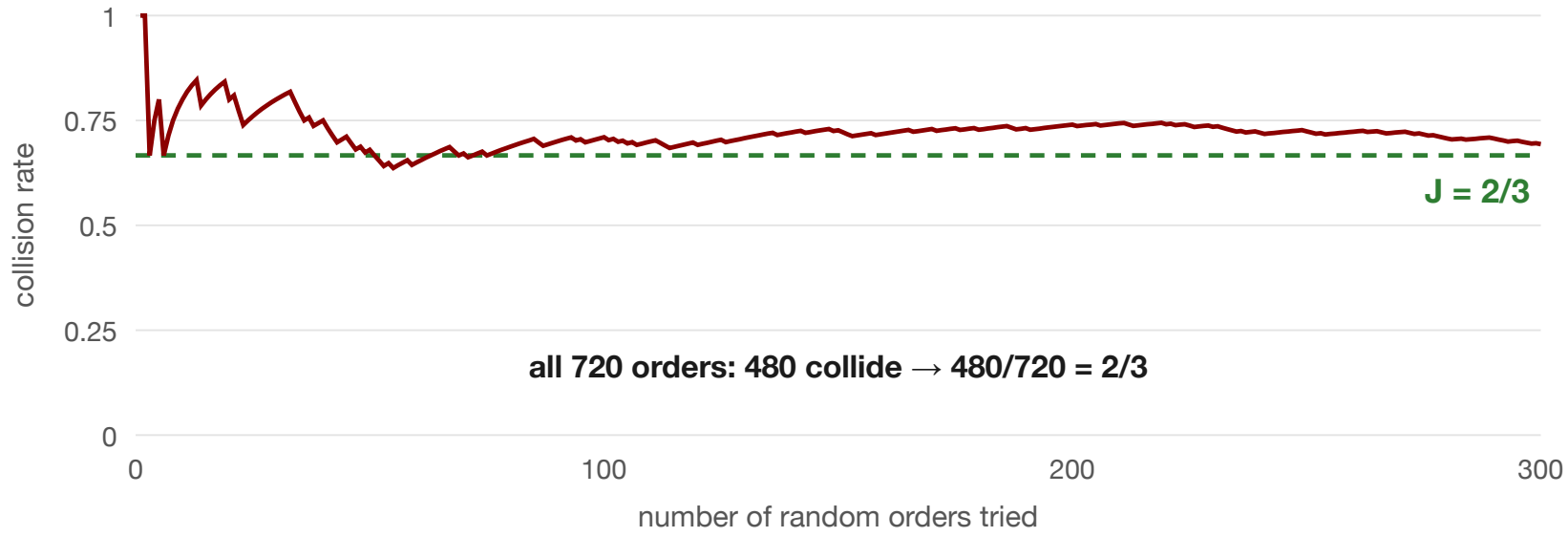
π_3	A	B
on the	**1**	**1**
cat sat	1	1
the sofa	0	1
the cat	1	1
sat on	1	1
the mat	1	0

collision

2 collisions out of 3 orders (J = 2/3)

Look at what decides a collision: the first row that either document has.

Tally: Is This a Coincidence?



*Not a coincidence. The collision probability **equals** the Jaccard similarity.*

The Claim

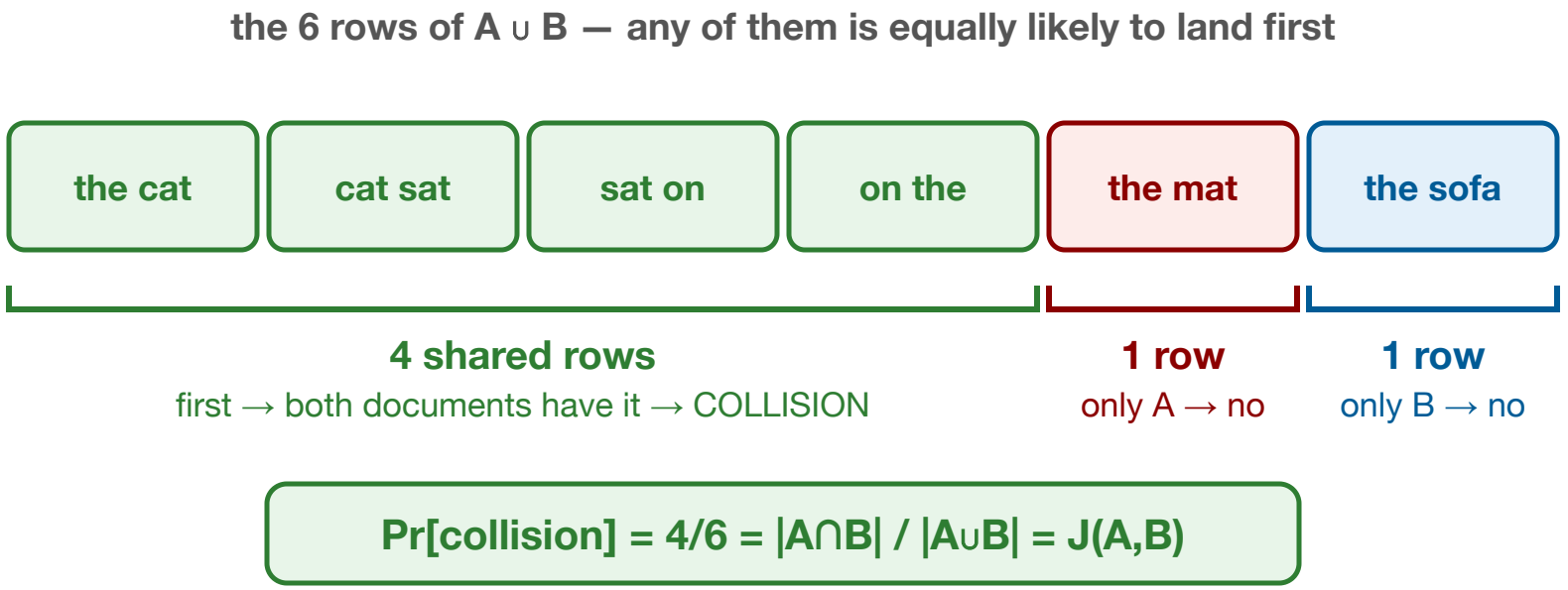
For a uniformly random order π of the rows,

$$\Pr\left[h_\pi(A) = h_\pi(B)\right] = J(A, B)$$

- This is exact: no approximation, no limit.
- One random order gives one yes/no outcome.
- The probability of "yes" is the similarity we want.

Next slide: why it is true, in one picture.

Why? The First Row of the Union Decides



Rows with $(0, 0)$ *never matter. Full proof, step by step: Appendix A.3.*

✅ Checkpoint 3 — MinHash

Test your understanding:

1. State the definition of $h_\pi(S)$ in one sentence.
2. For $A = \{1, 2, 3\}$ and $B = \{2, 3, 4, 5\}$, what is the probability that the MinHash values collide?
3. Why can rows with $(0, 0)$ be ignored?
4. Why is the collision probability exactly $|A \cap B|/|A \cup B|$?

Consolidate your answers before proceeding.

§ 4 — Signatures

Many fingerprints, a worked example, how accurate, how cheap

📎 *Extended material → Appendix A.4*

One Coin Flip Is Not Enough

A single MinHash is one yes/no outcome: a coin flip that lands "collide" with probability J. It says little.

Flip many coins. Use k independent random orders $\pi_1, \ldots, \pi_k$:

$$\mathrm{sig}(S) = \big(h_{\pi_1}(S), \ldots, h_{\pi_k}(S)\big) \qquad \hat{J}(A,B) = \frac{\#\text{positions where sig(A) and sig(B) agree}}{k}$$

Running example, $k = 3$ (orders π_1, π_2, π_3 above): sig(A) = (the mat, sat on, on the), sig(B) = (the cat, sat on, on the). They agree in 2 of 3 positions: $\hat{J} = 2/3$.

Permutations Are Too Expensive

- A shingle universe has about 10^7 rows. One stored permutation: 40 MB. For $k = 200$: **8 GB**, shared by every machine.

Use hash functions instead. A random function such as

$$h(x) = (a\,x + b) \bmod p, \qquad a, b \text{ random}$$

maps each row number x to a number. *Sorting rows by h plays the role of a random order.*

MinHash of a set: the **smallest** $h(x)$ over its rows.

A Worked Example: Four Documents, Five Shingles

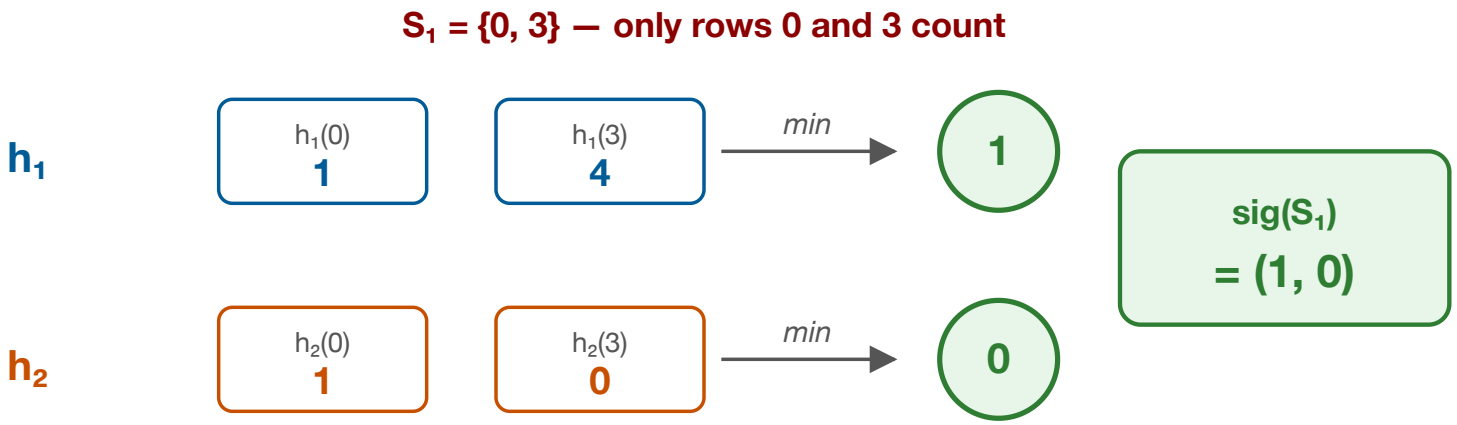
hash values: $h_1(x)=(x+1) \bmod 5$ $h_2(x)=(3x+1) \bmod 5$

x	S_1	S_2	S_3	S_4	$h_1(x)$	$h_2(x)$
0	**1**	0	0	**1**	1	1
1	0	0	**1**	0	2	4
2	0	**1**	0	**1**	3	2
3	**1**	0	**1**	**1**	4	0
4	0	0	**1**	0	0	3

5 shingles (rows), 4 documents (columns)

(Five shingles instead of six, so that we can do all the arithmetic by hand.)

Signature of One Document



Take the minimum of each hash function over the rows the document contains.

The Signature Matrix

characteristic matrix

	S_1	S_2	S_3	S_4
row 0	**1**	0	0	**1**
row 1	0	0	**1**	0
row 2	0	**1**	0	**1**
row 3	**1**	0	**1**	**1**
row 4	0	0	**1**	0

5 rows (millions in real data)



signature matrix

	S_1	S_2	S_3	S_4
h_1	**1**	**3**	**0**	**1**
h_2	**0**	**2**	**0**	**0**

k rows (k ≈ 200), any doc length

One column per document, k numbers per column, whatever the document length.

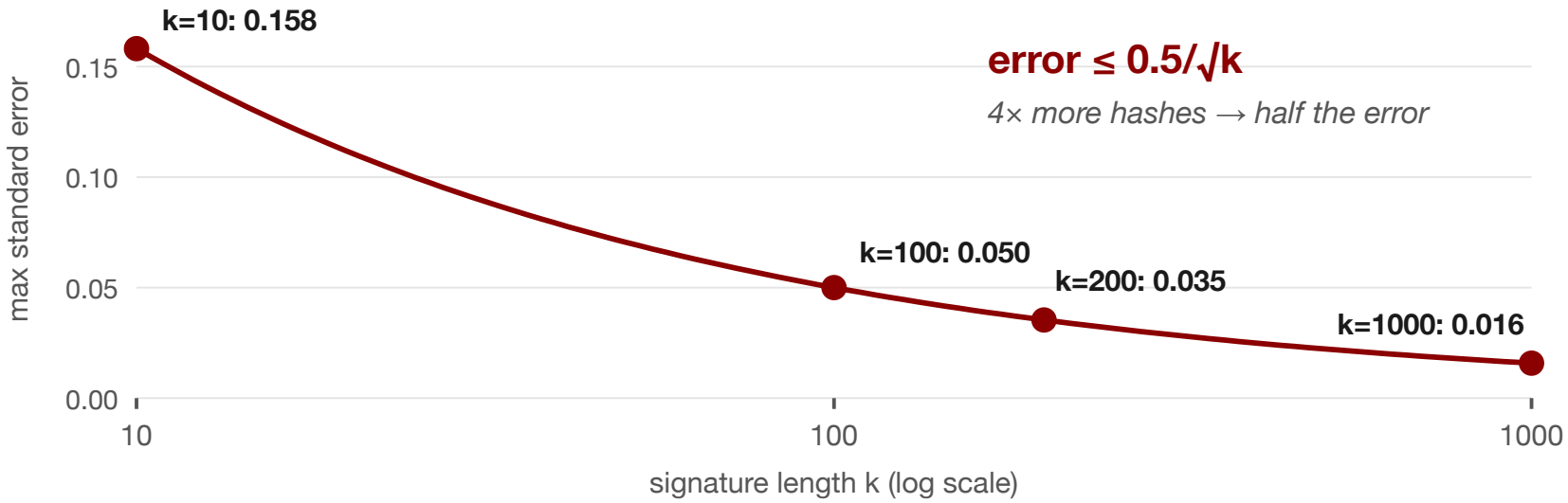
Estimating Similarity From Signatures

Fraction of agreeing positions versus the true Jaccard similarity:

pair	signatures	$\hat{J}$	true J
S_1, S_4	(1,0) vs (1,0)	1.00	0.67
S_1, S_3	(1,0) vs (0,0)	0.50	0.25
S_2, S_4	(3,2) vs (1,0)	0.00	0.33

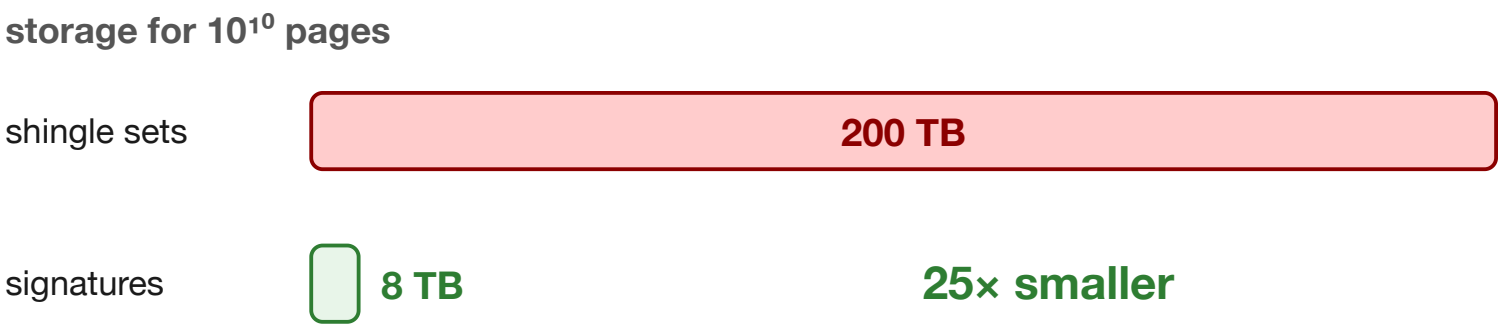
- The estimates are rough: we used only $k = 2$ hash functions.
- **More hash functions, better estimate.**

How Accurate? Error Shrinks as $1/\sqrt{k}$



- $\hat{J}$ is **unbiased**. For error ≤ 0.1 with probability 0.95: about $k = 185$ (the sampling bound of Lecture 3.4).

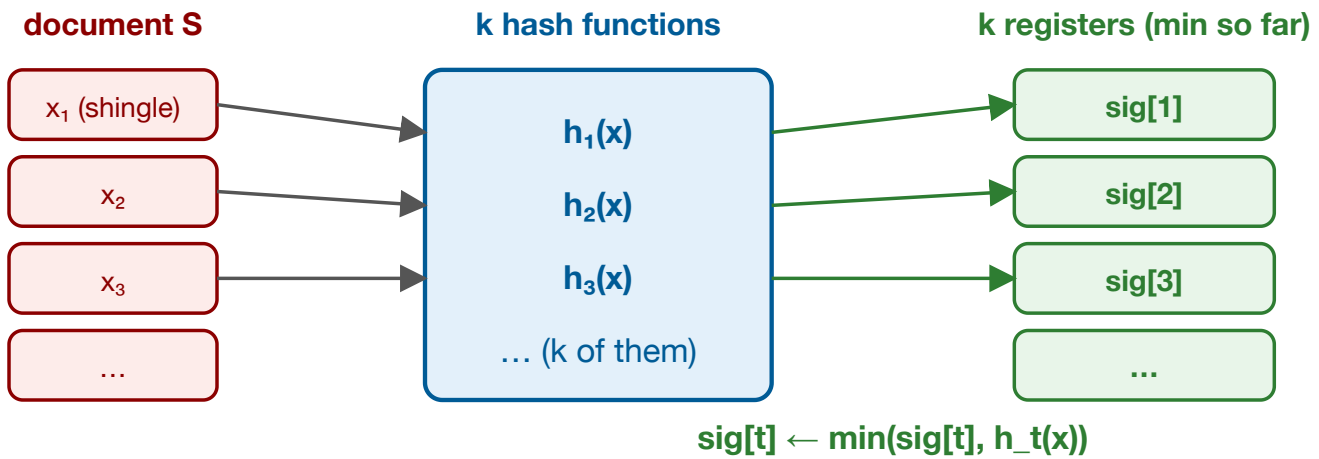
The Space Win



a 50-word page and a 50,000-word page: both get exactly 200 numbers

*The signature size does **not** depend on the length of the page.*

Computing Signatures in One Pass



one pass per document · documents are independent → a plain map, no shuffle

- Documents are independent: a plain `map` (Lectures 1.2, 1.3). **No shuffle.**
- Cost per document: (number of shingles) $\times$ k.

Pseudocode: Appendix A.4.1. Spark code: A.4.5.

✅ Checkpoint 4 — Signatures

Test your understanding:

1. Why is one MinHash value not enough to estimate J? How do we fix it?
2. Using $h_1(x) = (x + 1) \bmod 5$ and $h_2(x) = (3x + 1) \bmod 5$, compute the signature of $\{1, 2, 4\}$.
3. How does the standard error of $\hat{J}$ change if we quadruple k?
4. Why does the signature need only k numbers, however long the document?

Consolidate your answers before the synthesis.

§ 5 — Synthesis and Bridge

What we built, and what remains

Summary of Key Results

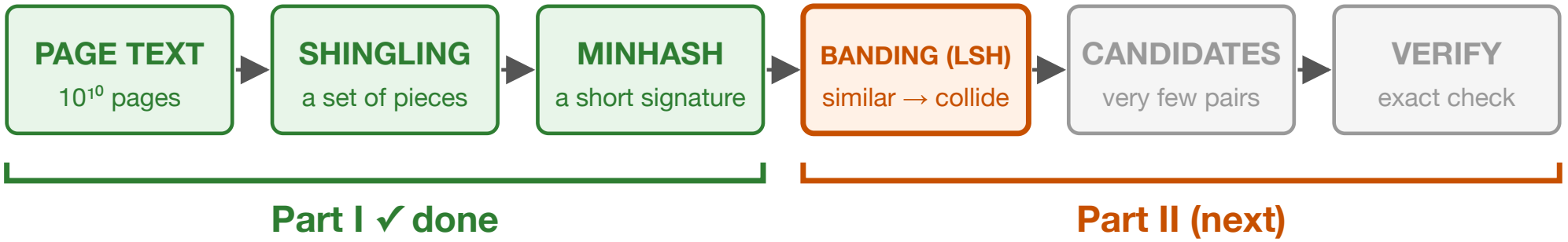
1. **Shingling** turns a document into a set; **Jaccard** $J = |A \cap B|/|A \cup B|$ measures similarity.
2. **MinHash:** under a random row order, $\Pr[h(A) = h(B)] = J(A, B)$.
3. A **signature** of k MinHash values estimates J with error $\leq 0.5/\sqrt{k}$: about 200 numbers per page.
4. **Hash functions** replace permutations; signatures are built in one parallel pass.

Common Student Confusions

- **Permutation vs. hash function.** A permutation is the ideal object in the proof. A hash function $h(x) = (ax + b) \bmod p$ is the cheap stand-in used in practice.
- **MinHash value vs. similarity.** A MinHash value is a *row*, not a similarity. The similarity is estimated by how often two documents' MinHash values **agree**.
- **Small signatures, still quadratic.** Signatures shrink each document, not the number of pairs.

The Remaining Problem, and the Idea for Part II

Still missing: avoid comparing all pairs of signatures



- 10^{10} signatures: **8 TB**, easy to store. But all pairs of signatures is still $\approx 5 \times 10^{19}$.
- **Idea:** hash the signatures so that **similar documents collide**; compare only those.

✅ Checkpoint 5 — Synthesis and Bridge

Test your understanding:

1. Name the three steps of Part I, and what each one shrinks.
2. What does a collision of two MinHash values tell us?
3. Which of the two obstacles of §2 have we removed, and which remains?
4. What would we like a hash function to do with similar documents?

Consolidate your answers; Part II starts from the remaining obstacle.

Lecture 3.3 (Part I) Complete

Recommended Reading

- Leskovec, Rajaraman & Ullman (2020). *Mining of Massive Datasets*, Ch. 3, §3.1–3.3.
- Broder (1997). *On the resemblance and containment of documents*. Compression and Complexity of Sequences (SEQUENCES).

Contact & Q&A

- **Course:** Big Data Computing
- **Office Hours:** Drop me a message, and we'll arrange a time to meet, either in person or online! 🙂

Thank you for your attention!

§ Appendix

Appendix Map

Section	Extends	What you will find	Slides
A.1	§ 1	Pair-counting and all-pairs cost arithmetic	1
A.2	§ 2	Jaccard distance is a metric, choosing k, containment and size imbalance, shingling variants	4
A.3	§ 3	The MinHash theorem, full proof step by step, and sanity checks on other pairs	3
A.4	§ 4	Signature pseudocode, hash families in practice, variance of $\hat{J}$, signature length table, MinHash on Spark	5
A.5	§ 4	Exam-style worked problems with answers	2

Supplementary material for deeper study. Not required during the lecture.

📎 A.1 — Scale Arithmetic

A.1.1 · Counting Pairs and All-Pairs Cost

$\binom{n}{2} = n(n-1)/2 \approx n^2/2.$

n	pairs	time at 10^{12} comparisons/s
10^6	5×10^{11}	0.5 s
10^9	5×10^{17}	≈ 5.8 days
10^{10}	5×10^{19}	≈ 1.6 years

- Pairs grow **quadratically**: $10\times$ more pages means $100\times$ more pairs.
- One comparison of two raw pages costs thousands of operations, so these times are lower bounds.

📎 A.2 — Sets, Jaccard and Shingles

A.2.1 · Jaccard Distance Is a Metric

Define $d_J(A, B) = 1 - J(A, B)$. Symmetry, $d_J \geq 0$ and $d_J(A, A) = 0$ are immediate. The triangle inequality follows from MinHash:

- By §3, $d_J(A, B) = \Pr_\pi[h_\pi(A) \neq h_\pi(B)]$.
- If $h_\pi(A) \neq h_\pi(B)$, then $h_\pi(A) \neq h_\pi(C)$ or $h_\pi(C) \neq h_\pi(B)$ (else they would be equal).
- Hence $\Pr[h(A) \neq h(B)] \leq \Pr[h(A) \neq h(C)] + \Pr[h(C) \neq h(B)]$, that is

$$d_J(A, B) \leq d_J(A, C) + d_J(C, B) \qquad \blacksquare$$

A.2.2 · Choosing the Shingle Length k

With 27 symbols (letters and space), the number of possible character shingles is 27^k:

k	possible shingles
5	$27^5 \approx 1.4 \times 10^7$
9	$27^9 \approx 7.6 \times 10^{12}$

- A typical page has a few thousand shingles. With $k = 5$ it fills a visible share of 1.4×10^7 possible shingles; with $k = 9$ almost none.
- So $k = 9$ makes a shared shingle **informative**: unrelated pages almost never share one.
- Long shingles are hashed to 32 or 64 bits ($2^{32} \approx 4.3 \times 10^9$) to get fixed-size integer ids.

A.2.3 · Size Imbalance and Containment

Jaccard punishes **size differences**. Let $A \subset B$ with $|A| = 10$ and $|B| = 1{,}000$:

$$J(A, B) = \frac{10}{1000} = 0.01, \qquad \text{but } A \text{ is entirely inside } B$$

Containment $c(A, B) = |A \cap B|/|A| = 1$ captures "A is part of B".

- Jaccard is right for *near-duplicates* (similar size by definition).
- For "is this snippet quoted in that page?", containment is the better question. Broder (1997) treats both.

A.2.4 · Shingling Variants

- **Character shingles:** robust to small typos, many shingles per page.
- **Word shingles** ($k = 2$ or 3 words): fewer, more meaningful shingles; sensitive to word-level edits.
- **Preprocessing:** lowercase, strip markup and punctuation, collapse whitespace, before shingling.
- **Boilerplate:** menus and footers appear on all pages of a site and inflate J. Remove them, or shingle only the main text.
- **Sets, not bags:** repeated shingles count once.

📎 A.3 — The MinHash Theorem, Step by Step

A.3.1 · Proof (1 of 2): Reducing to One Row

Setup. Rows of the characteristic matrix; A, B sets of rows; π a uniformly random order.

1. Let $U = A \cup B$, $|U| = u$. For $S \in \{A, B\}$, $h_\pi(S)$ is the first row of S in the order π. Because $S \subseteq U$, this is the first row of S **within** U.
2. Let x^* be the first row of U under π. By symmetry of a uniform random order, $\Pr[x^* = x] = 1/u$ for every $x \in U$.
3. x^* belongs to A or to B (or both), because it is in U.

A.3.2 · Proof (2 of 2): When Do They Collide?

4. If $x^* \in A \cap B$: it is the first row of A and also the first of B, so $h_\pi(A) = h_\pi(B) = x^*$. **Collision.**
5. If $x^* \in A \setminus B$: $h_\pi(A) = x^*$, but B does not contain x^*, so $h_\pi(B) \neq x^*$. **No collision.** The case $x^* \in B \setminus A$ is symmetric.
6. Therefore $h_\pi(A) = h_\pi(B) \iff x^* \in A \cap B$, and

$$\Pr[h_\pi(A) = h_\pi(B)] = \sum_{x \in A \cap B} \Pr[x^* = x] = \frac{|A \cap B|}{|A \cup B|} \qquad \blacksquare$$

Check with the running example: $|U| = 6$ rows, 4 of them shared. Over all 720 orders, 480 collide: $480/720 = 4/6$.

A.3.3 · Sanity Check: The Same Rule on Other Pairs

pair	$\lvert A \cap B \rvert$	$\lvert A \cup B \rvert$	collision probability
running example	4	6	$4/6 = 0.67$
identical sets	n	n	1
disjoint sets	0	$n_A + n_B$	0
$A = \{1,2,3,4\}$, $B = \{3,4,5\}$	2	5	$2/5 = 0.4$

The rule agrees with common sense at both extremes.

📎 A.4 — Hash Families, Variance, Signature Length, Spark

A.4.1 · Signature Computation, in Pseudocode

```
sig[S][t] = +infinity   for every document S and t = 1..k
for each document S:                        # independent of other documents
    for each shingle x in S:
        for t = 1..k:
            sig[S][t] = min(sig[S][t], h_t(x))
```

- Documents are processed **independently**: a plain `map` over documents (Lectures 1.2, 1.3).
- **No shuffle** is needed to build signatures.
- Cost: (number of shingles) $\times$ k operations per document.

Spark version: A.4.5.

A.4.2 · Hash Functions in Practice

A standard family: $h_{a,b}(x) = (a\,x + b) \bmod p$, with p a prime larger than the universe of row ids, $a \in \{1, \ldots, p-1\}$, $b \in \{0, \ldots, p-1\}$ random.

- It is **2-universal**: any two rows collide with probability about $1/p$.
- It is **not exactly min-wise independent**, so the collision probability is only approximately J. In practice the error is negligible for good families.
- Exactly min-wise independent families exist (Broder, Charikar, Frieze & Mitzenmacher, 1998) but are costlier.
- Variants that save space: keep only a few bits of each MinHash value (Li & König, 2010).

A.4.3 · Variance of the Estimator

Each position t gives an indicator $X_t \in \{0, 1\}$ with $\Pr[X_t = 1] = J$, independent across t. Then $\hat{J} = \frac{1}{k}\sum_t X_t$ and:

$$E[\hat{J}] = J, \qquad \mathrm{Var}(\hat{J}) = \frac{J(1-J)}{k} \le \frac{1}{4k}, \qquad \mathrm{SE} \le \frac{0.5}{\sqrt{k}}$$

- Worst case $J = 0.5$: $k = 200$ gives $\sqrt{0.25/200} \approx 0.035$.
- At $J = 2/3$: $\sqrt{(2/3)(1/3)/200} \approx 0.033$.
- Quadrupling k halves the error: the $1/\sqrt{k}$ law of sampling.

A.4.4 · How Long Must the Signature Be?

Hoeffding (derived in Lecture 3.4): $\Pr[|\hat{J} - J| \geq \epsilon] \leq 2e^{-2k\epsilon^2}$. Setting the right side to δ:

$$k \geq \frac{\ln(2/\delta)}{2\epsilon^2}$$

error ϵ	confidence $1-\delta$	k
0.10	0.95	185
0.10	0.99	265
0.05	0.95	738
0.05	0.99	1,060

Halving the error multiplies k by about 4.

A.4.5 · MinHash Signatures on Spark

```
import numpy as np
P = 2**31 - 1                          # prime larger than any shingle id
K = 200                                # signature length
rng = np.random.default_rng(7)
A = rng.integers(1, P, size=K, dtype=np.int64)   # hash parameters, shared
B = rng.integers(0, P, size=K, dtype=np.int64)

def signature(shingle_ids):            # one document: a set of ints in [0, P)
    x = np.fromiter(shingle_ids, dtype=np.int64)
    h = (np.outer(A, x) + B[:, None]) % P        # K hashes of every shingle
    return h.min(axis=1)                         # min over the set: K numbers

# docs: RDD of (doc_id, set of shingle ids)   ->   map-only, no shuffle
sigs = docs.mapValues(signature)
```

The parameters `A`, `B` are broadcast once. Products stay below 2^{62}, so no overflow.

📎 A.5 — Exam-Style Worked Problems

A.5.1 · Problems 1–2

P1. Compute the character 2-shingles of $x =$ "abcab" and $y =$ "abcbc", and $J(x, y)$.

Answer. $\{ab, bc, ca\}$ and $\{ab, bc, cb\}$: $|\cap| = 2$, $|\cup| = 4$, $J = 0.5$.

P2. Rows $0..5$; $X = \{0, 2, 5\}$, $Y = \{2, 3, 5\}$; $h_1(x) = (x + 3) \bmod 6$, $h_2(x) = (5x + 2) \bmod 6$. Compute both signatures and $\hat{J}$.

Answer. h_1 values on rows $0..5$: $3, 4, 5, 0, 1, 2$; h_2: $2, 1, 0, 5, 4, 3$. $\text{sig}(X) = (2, 0)$, $\text{sig}(Y) = (0, 0)$, so $\hat{J} = 1/2$ (true $J = 2/4 = 0.5$).

A.5.2 · Problems 3–4

P3. How many hash functions guarantee $|\hat{J} - J| \leq 0.05$ with probability ≥ 0.95?

Answer. $k \geq \ln(2/0.05)/(2 \cdot 0.05^2) = 3.689/0.005 \approx 738$.

P4. $|A| = 100$, $|B| = 50$, $|A \cap B| = 40$. Give $J(A, B)$ and the probability that one MinHash collides.

Answer. $|A \cup B| = 100 + 50 - 40 = 110$, so $J = 40/110 \approx 0.364$, and that is the collision probability.

P5. $|A| = 10 \subset B$, $|B| = 1000$. Give J and the containment $c(A, B)$.

Answer. $J = 0.01$, $c(A, B) = 1$.