

INFORMATION RETRIEVAL

Week 4 – Tolerant Retrieval

Today

1

Exercise Recap

- Discussion
- Questions

2

Theory

- B+-Trees
- Wildcard Queries
- Permuterm Index
- Spelling correction

3

Bonus

- Exercise 3: Tolerant Retrieval

4

Kahoot / Exam questions

21.03.2025

Exercise 2

Discussion

	Wahr	Falsch	
	☐	☐	Stemming should be invoked at indexing time but not while processing a query.
	☐	☐	In a Boolean retrieval system, stemming never lowers recall.
	☐	☐	Stemming increases the size of the vocabulary.
	☐	☐	In a Boolean retrieval system, stemming never lowers precision.

Exercise 2

Discussion

- Which of the following bit strings contain only valid UTF-8 encoded data?
 - 01001000

Yes it does, 8-bit string starting with 0

Exercise 2

Discussion

- Which of the following bit strings contain only valid UTF-8 encoded data?
 - 11001110 10110001

Yes it does, two 8-bit strings, first one starts with “110” signaling that the following 8-bit block also belongs to the first and needs to start with “10”.

Exercise 2

Discussion

- Which of the following bit strings contain only valid UTF-8 encoded data?
 - 11000000 0100000

No, since the first 8-bit string starts with “110”, signaling that the next block also belongs to it, but the next block does not start with “10”.

Exercise 2

Discussion

- Which of the following bit strings contain only valid UTF-8 encoded data?
 - 10101010 01011001

No, since the first block starts with “10”.

Exercise 2

Discussion

- Which of the following bit strings contain only valid UTF-8 encoded data?
 - 0111

No, only 4 bits.

Exercise 2

Discussion

- Which of the following bit strings contain only valid UTF-8 encoded data?
 - 01001010 01000001

Yes, two 8-bit strings, both starting with 0.

Exercise 2

Discussion

- Which of the following bit strings contain only valid UTF-8 encoded data?
 - 11100010 10011101 10100100

Yes, first string starts with “1110”, indicating the next two blocks belong to it. The next two blocks both start with “10”.

Exercise 2

Questions?

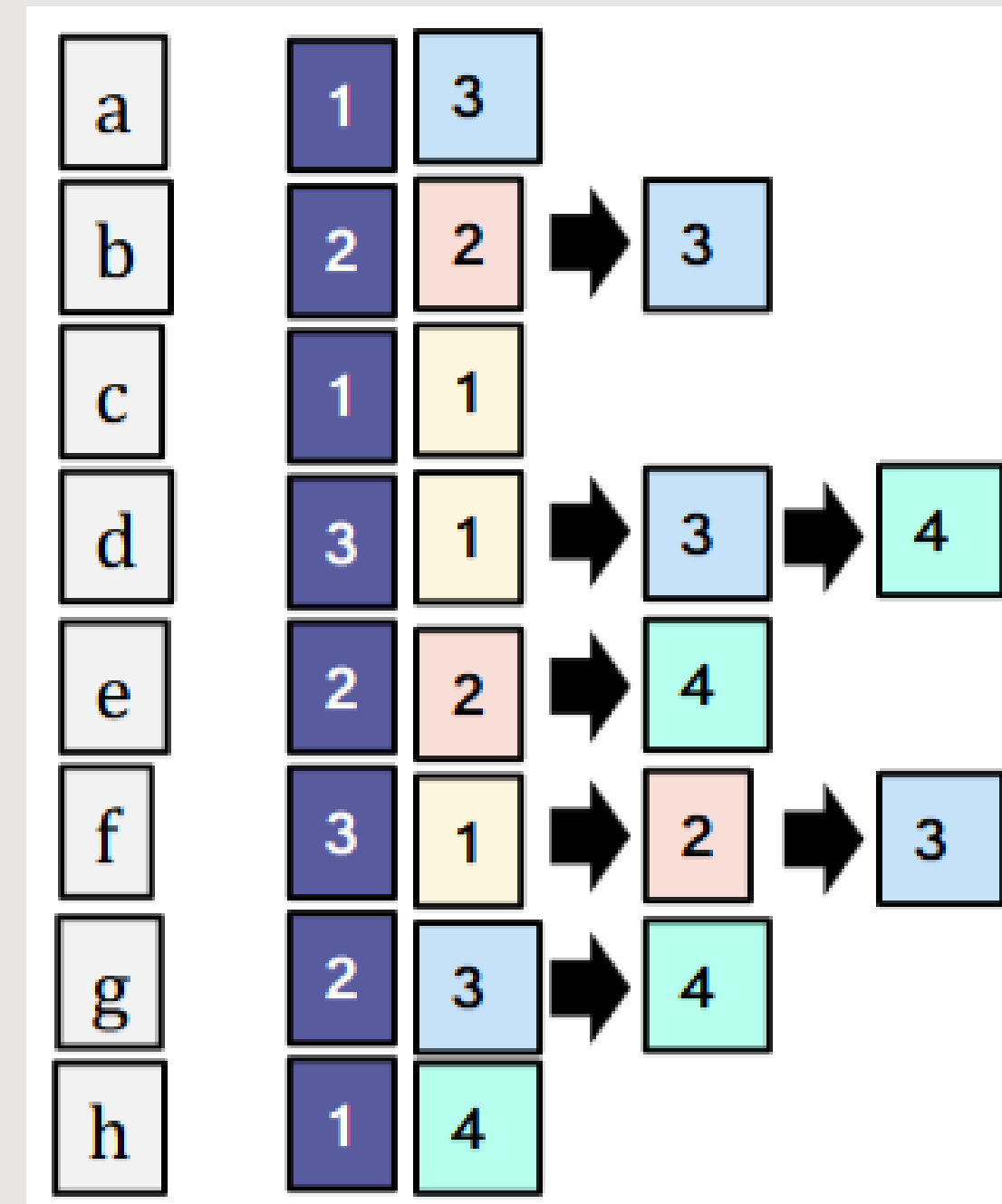
21.03.2025

Tolerant Retrieval

B⁺-Trees

- Issues/Limitations with hash tables?

Range Queries, Space Requirements



B+-Trees

Solution: Binary Search Trees

Even better: B+-trees

- Taking advantage of block accesses

Rules:

- All leaves at same depth
- Terms / postings lists are only at the leaves
- If #keys between d and $2d$, then #children between $d + 1$ and $2d + 1$
- Root can have less children

1-1 B+-tree

2-3 B+-tree

3-5 B+-tree

4-7 B+-tree

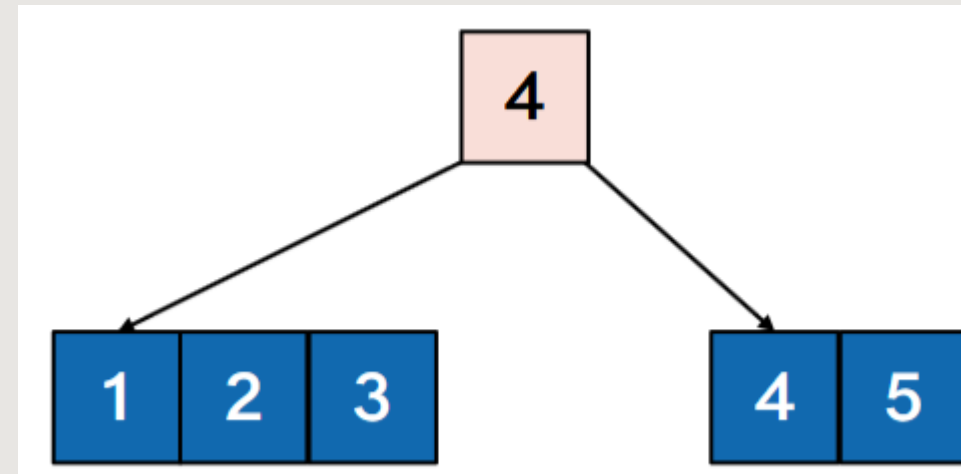
3-5 B+-Tree



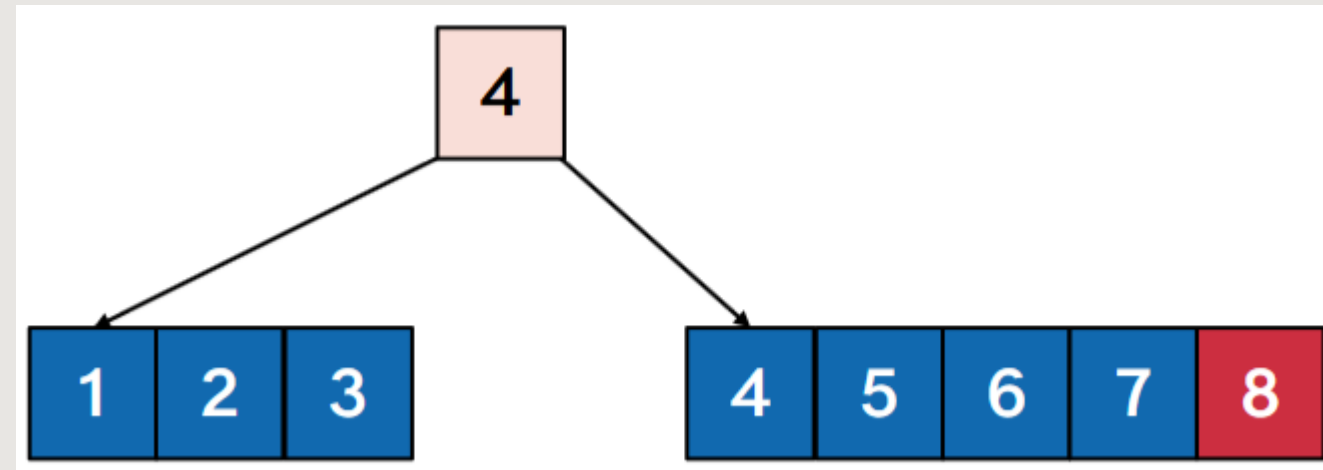
3-5 B+-Tree



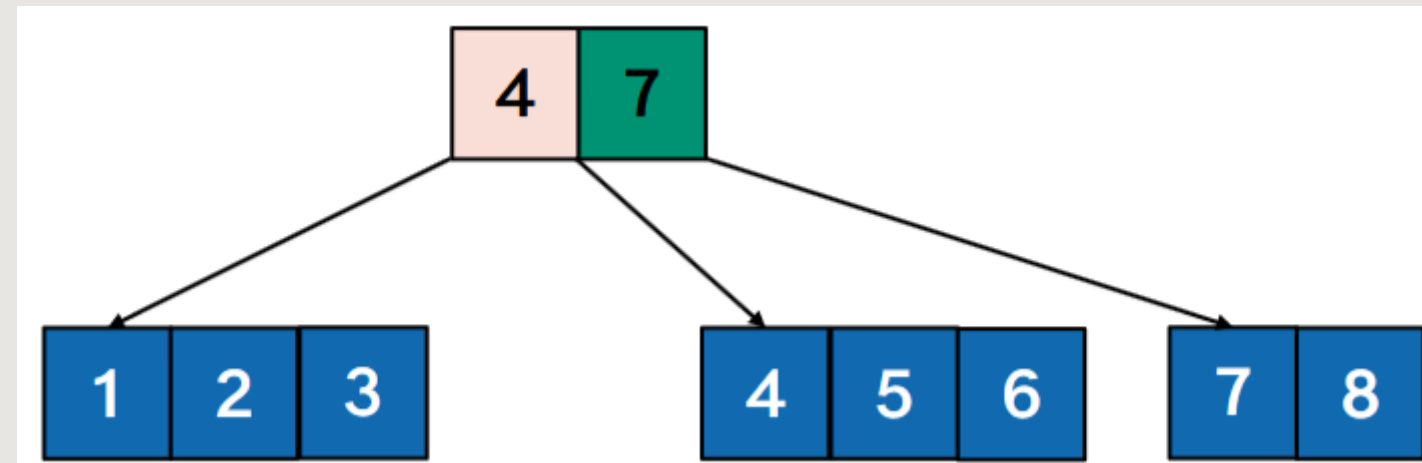
3-5 B+-Tree



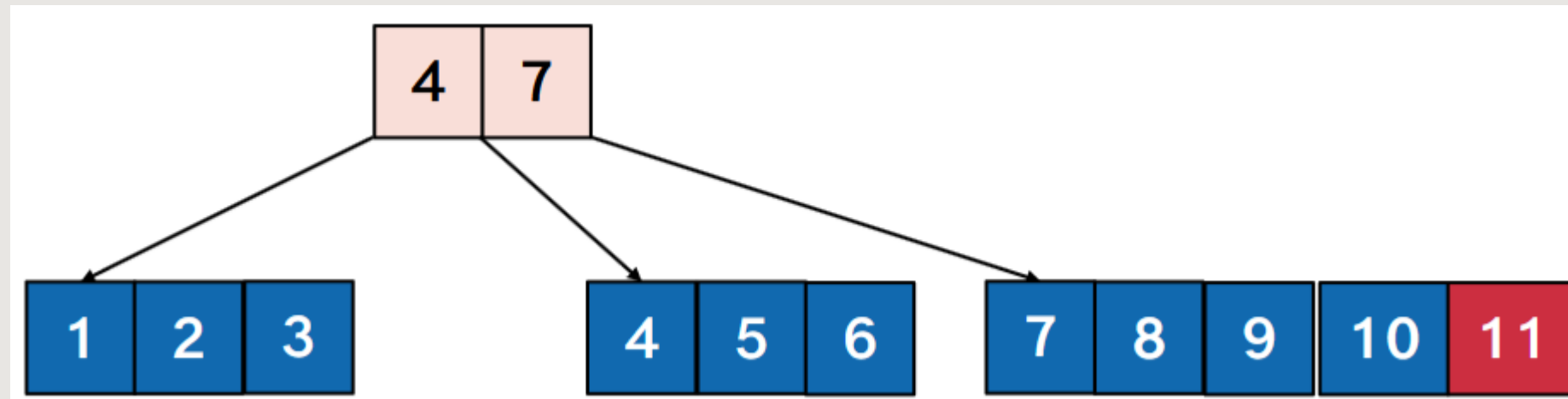
3-5 B+-Tree



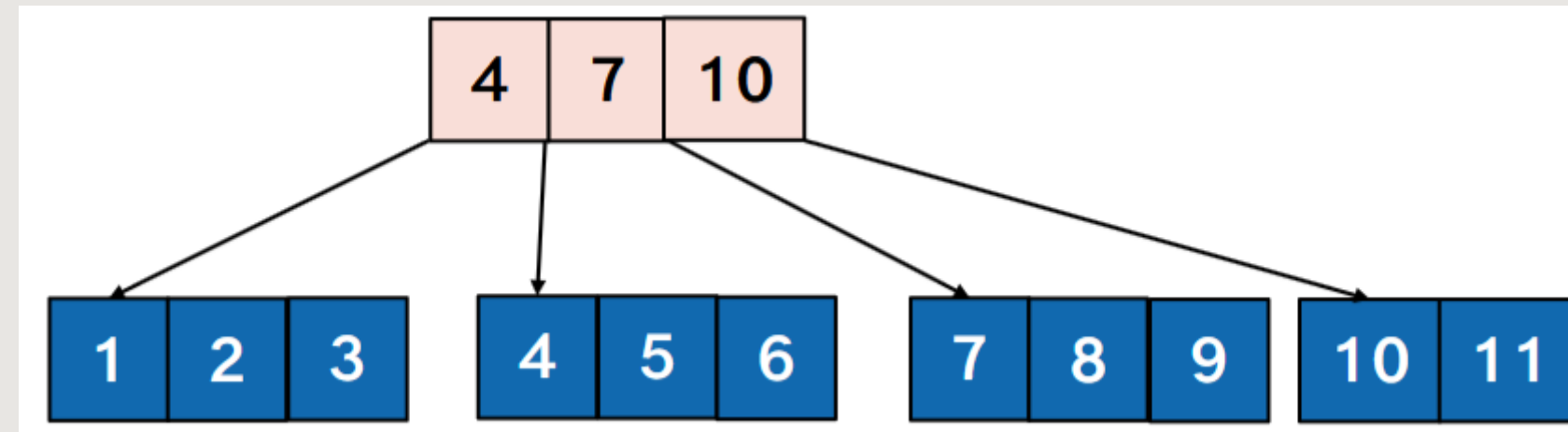
3-5 B+-Tree



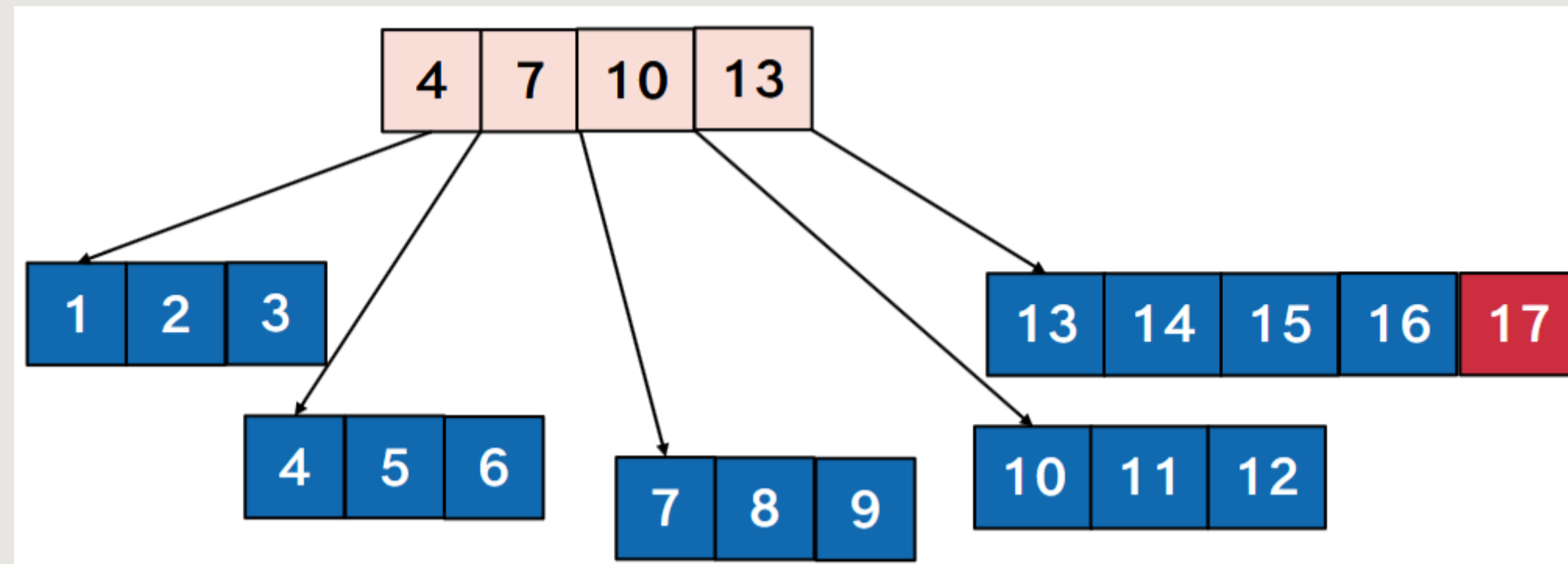
3-5 B+-Tree



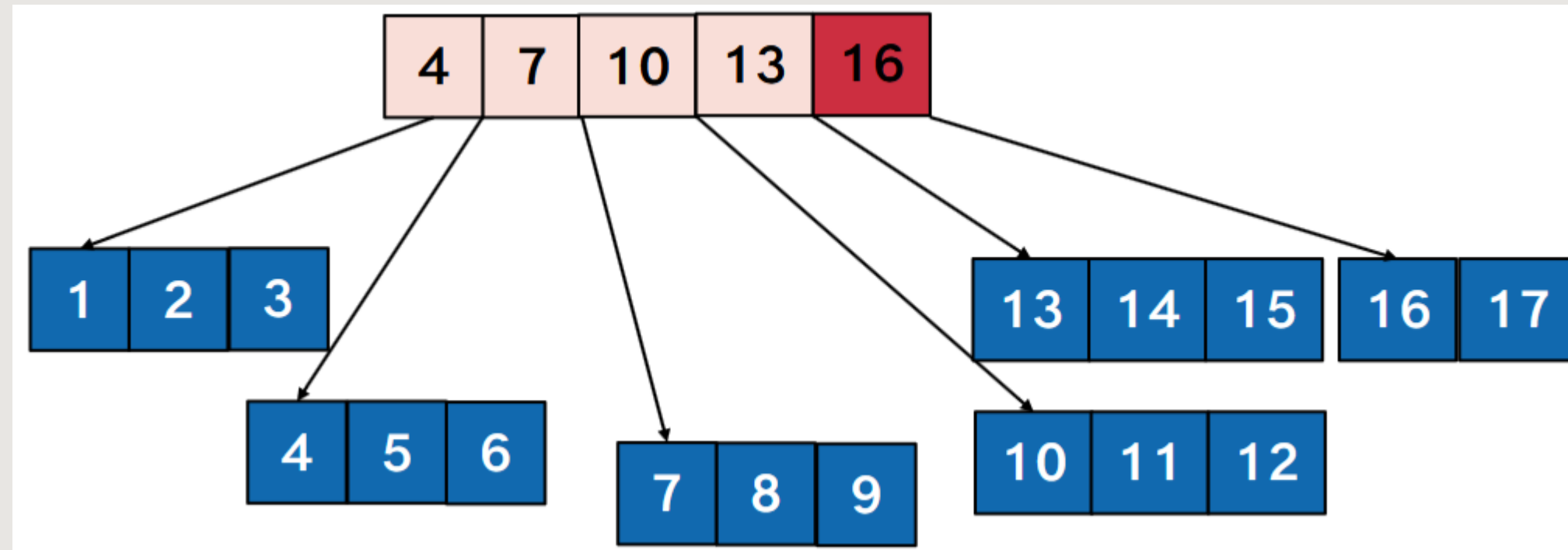
3-5 B+-Tree



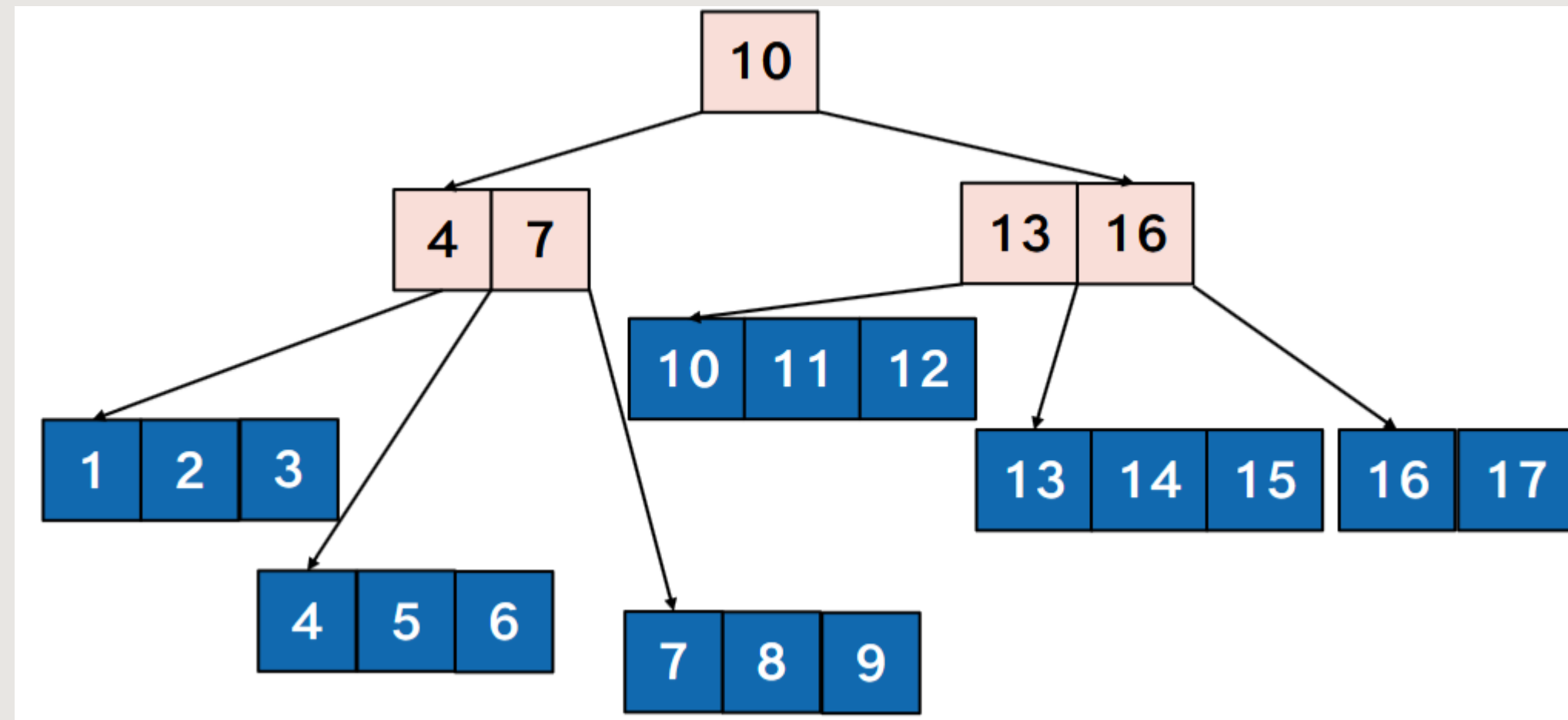
3-5 B+-Tree



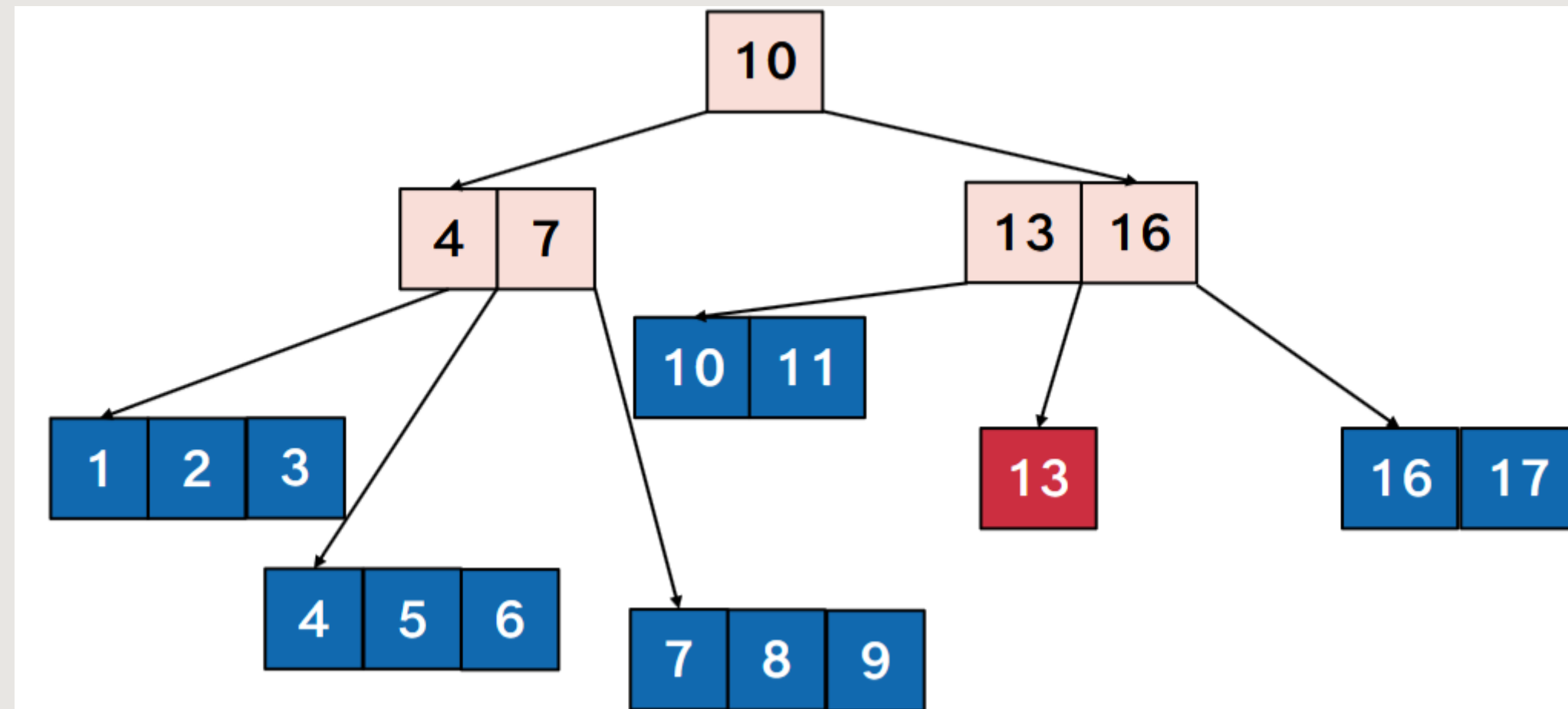
3-5 B+-Tree



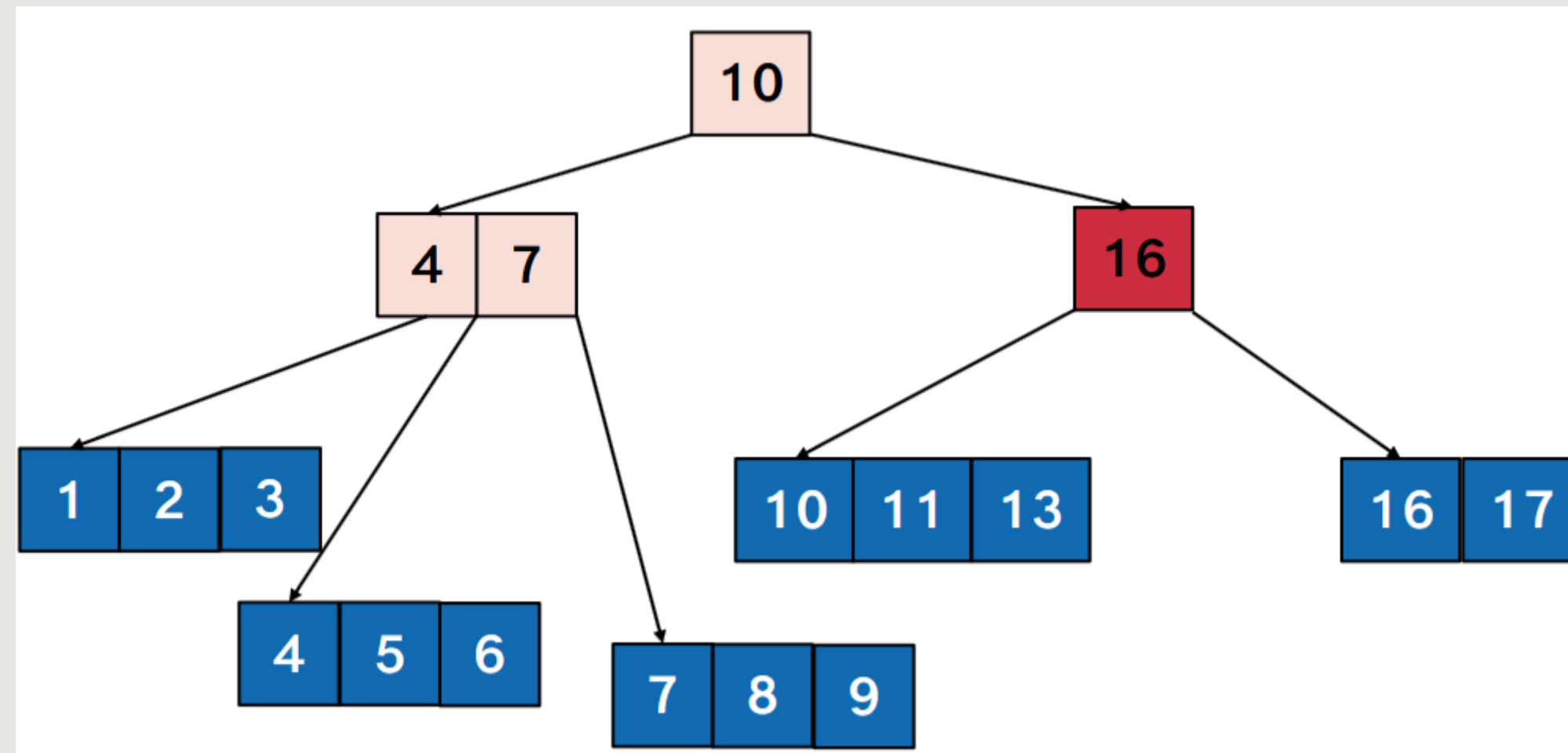
3-5 B+-Tree



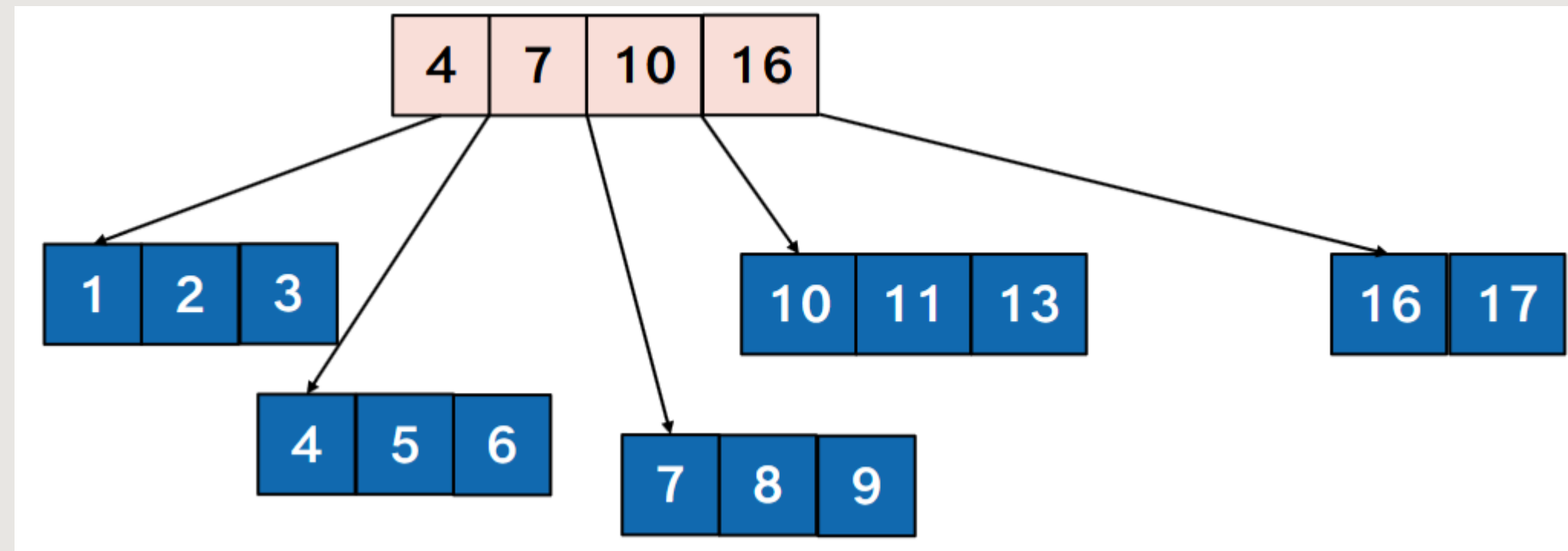
3-5 B+-Tree



3-5 B+-Tree



3-5 B+-Tree

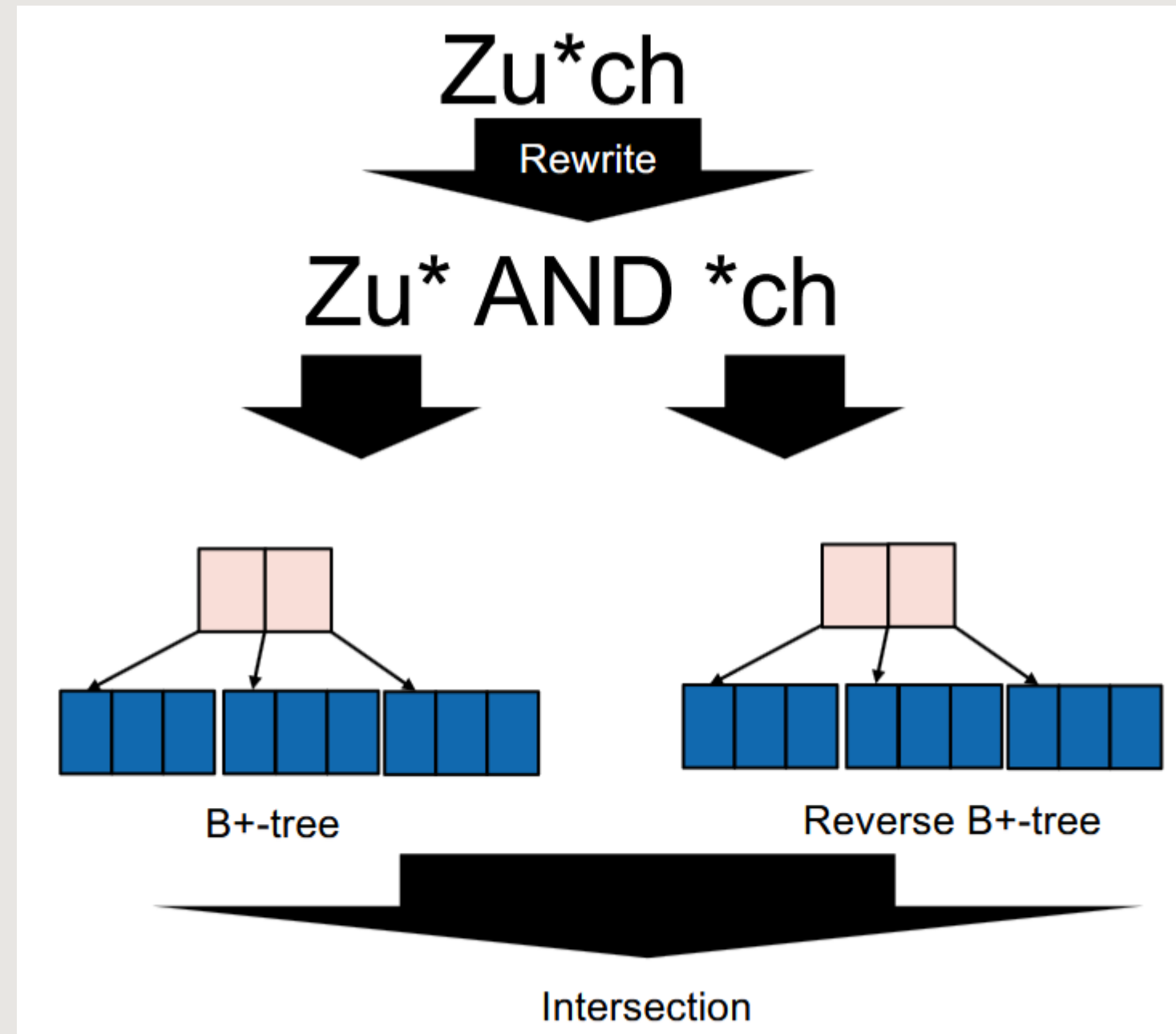


Wildcard queries

- e.g. “Z*rich”
 - Allows different spellings, families of words, etc.
- Leading vs. Trailing queries (“*ics” vs. “Math*”)
 - “Math*”, just find all matches for Math in the B+-tree
 - “*ics”, reverse B+-tree, then do the same

Naïve approach

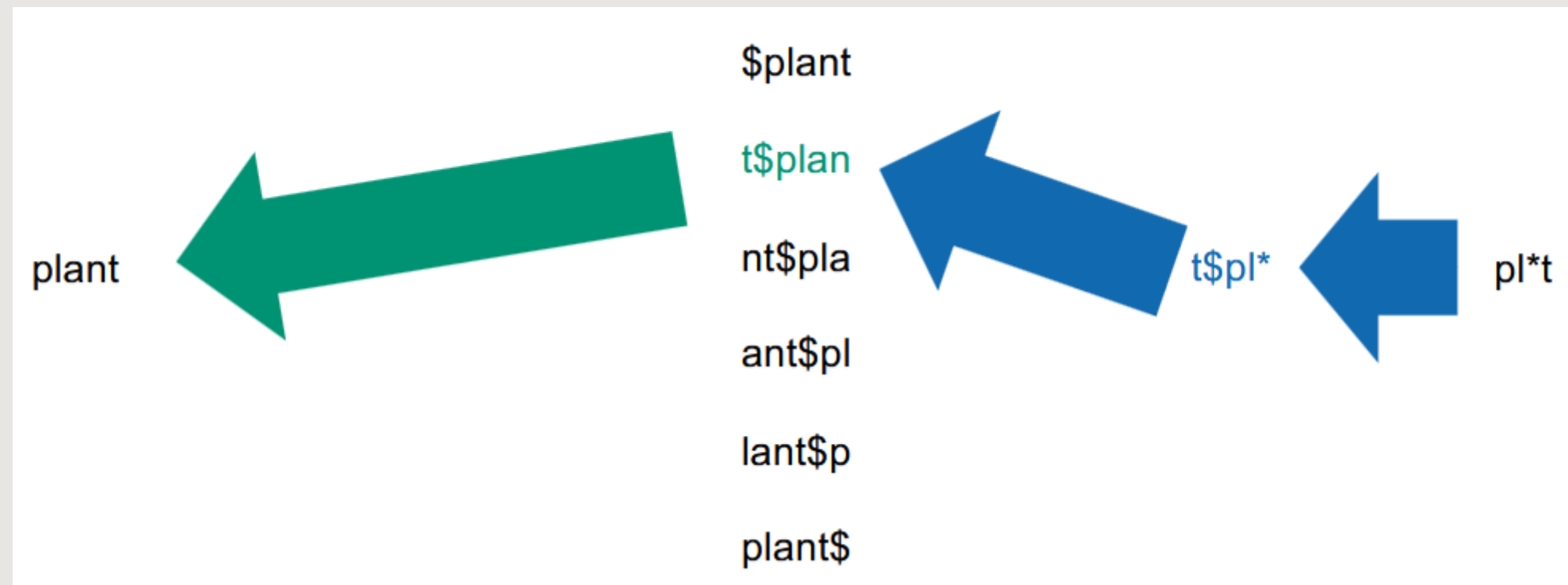
- Large Storage usage
- Inefficient
- False positives



Permuterm index



Permuterm index

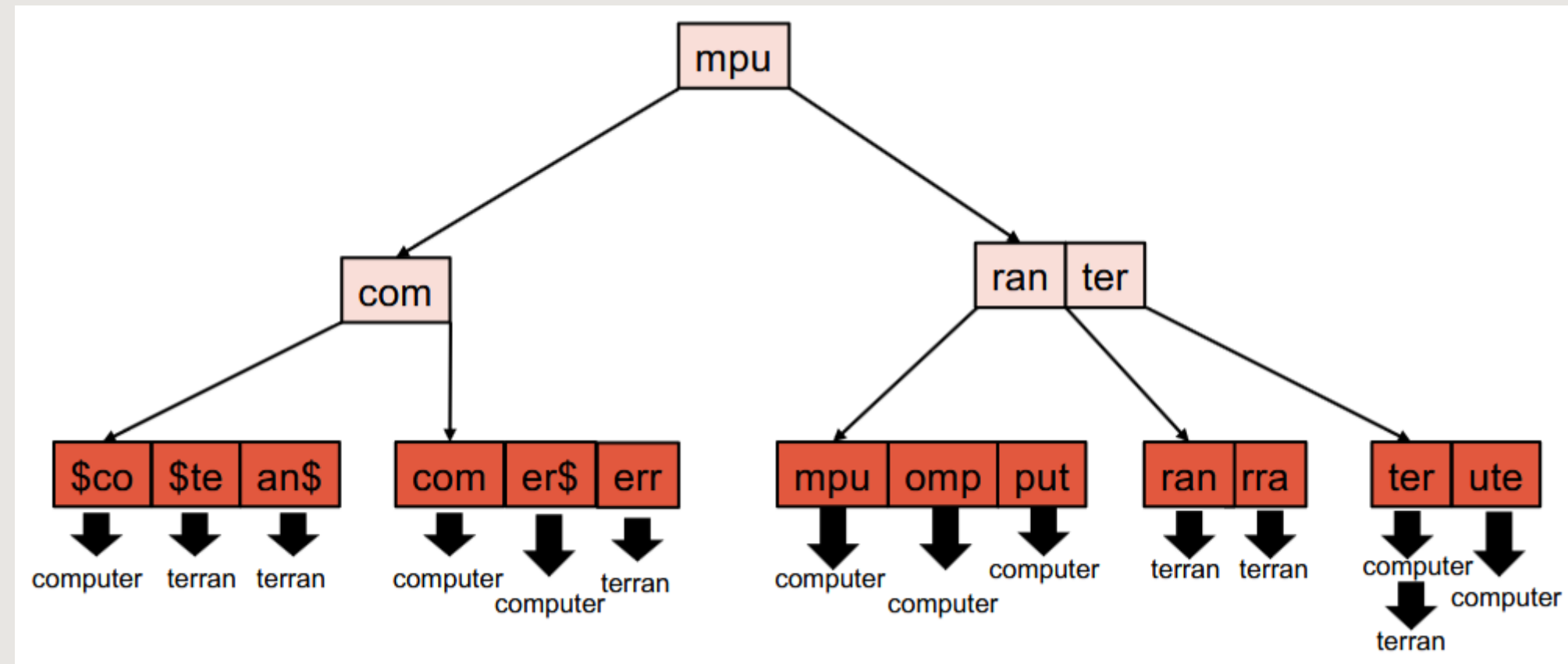


k-gram index

computer	1-grams	\$, c, o, m, p, u, t, e, r, \$	Not very useful
	2-grams	\$c, co, om, mp, pu, ut, te, er, r\$	
	3-grams	\$co, com, omp, mpu, put, ute, ter, er\$	
	4-grams	\$com, comp, ompu, mput, pute, uter, ter\$	
	5-grams	\$comp, compu, omput, mpute, puter, uter\$	
	6-grams	\$compu, comput, ompute, mputer, puter\$	Not space efficient
	7-grams	\$comput, compute, omputer, mputer\$	
...			

Wildcard Queries

k-gram index



Edit / Levenshtein distance (DP...)

	#	a	t	e
#	0	1	2	3
c	1	1	2	3
a	2			
t	3			

Diagram illustrating the calculation of the Edit / Levenshtein distance using Dynamic Programming (DP).

The table shows the distance between the prefix of the source word "cat" (rows) and the prefix of the target word "ate" (columns). The cells are colored blue for the header, green for the source word characters, and light gray for the target word characters.

Annotations for the calculation of the distance for the prefix "ca" (row 3, column 2):

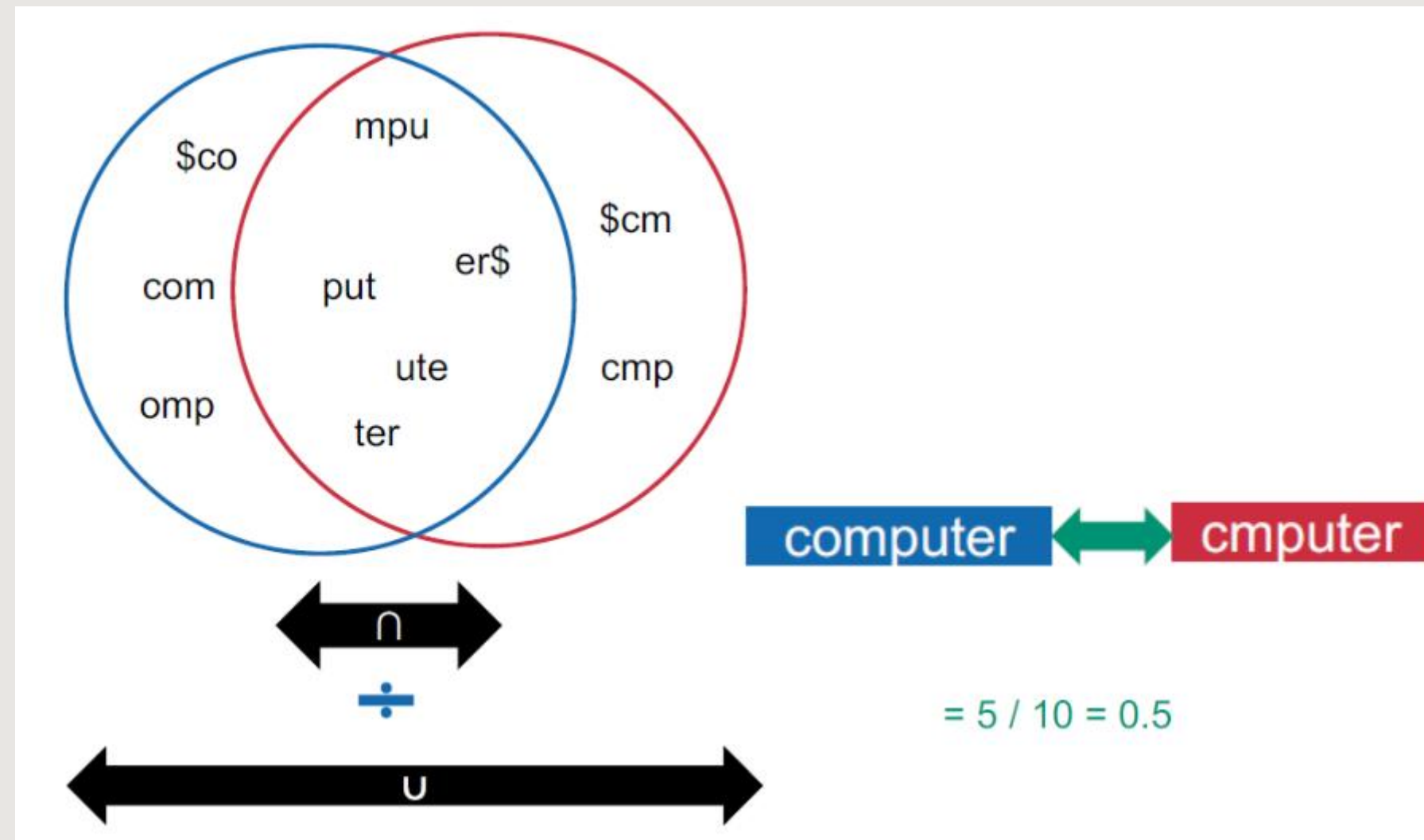
- From cell (3, 1) to (3, 2): 0 (do nothing)
- From cell (2, 2) to (3, 2): $+1$ (add a)
- From cell (2, 3) to (3, 2): $+1$ (remove a)
- The value in cell (3, 2) is $\min(3, 1, 2)$.

Spelling Correction

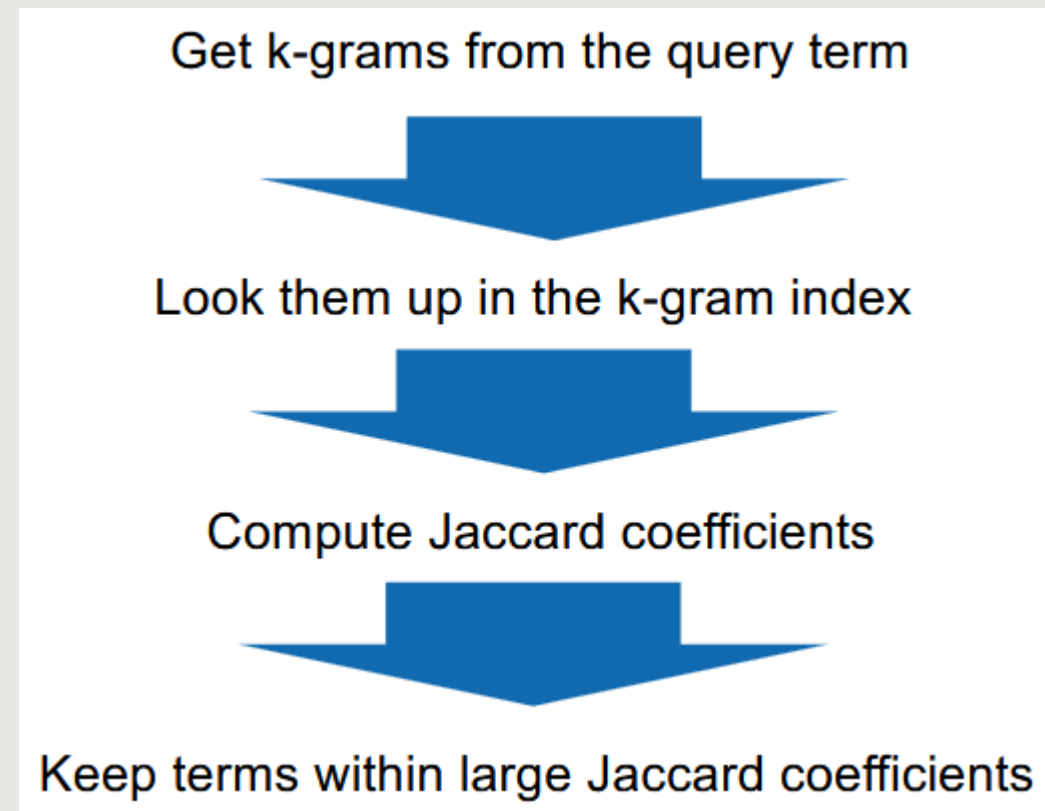
Edit distance (DP...)

- Issue: Costly computations if done on a lot of terms
- Solution: Preselect a few terms
- How?

Jaccard coefficient



Jaccard coefficient



Soundex algorithm

Change...	To...
A E H I O U W Y	0
B F P V	1
C G J K Q S X Z	2
D T	3
L	4
M N	5
R	6

Computer	➡	C0510306	➡	C0510306	➡	C5136	➡	C513
Cmputer	➡	C510306	➡	C510306	➡	C5136	➡	C513
Zurich	➡	Z06020	➡	Z06020	➡	Z62	➡	Z620
Information	➡	I510650305	➡	I510650305	➡	I516535	➡	I516
Letter	➡	L03306	➡	L0306	➡	L36	➡	L360
Retrieval	➡	R0360104	➡	R0360104	➡	R3614	➡	R361

Exercise 3: Tolerant Retrieval

Bonus Quiz

- Moodle-based
- Start: 21.03 at 15:00
- Deadline: 28.03 at 15:00
- 8 / 16 to pass
- Last two questions involve programming exercises
- The rest are theoretical questions
 - Edit distance
 - Jaccard coefficient
 - B+-trees
- **Two attempts** in total

21.03.2025

Programming exercise

- Implement k-gram index
- Process wildcard queries (Bonus)
- Process spell correction (Bonus)

Programming exercise

Wildcard queries

- Implement `wildcard_parse(q)`
- Takes wildcard query (“som*e”) and returns list containing all keys to query the k-gram index

```
1 def wildcard_parse(q):  
2     ### TODO for the assignment: parse the wildcard term and return a list of k-grams  
3     return []
```

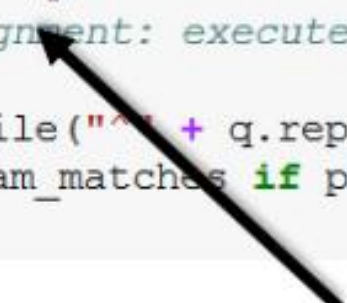


Programming exercise

Wildcard queries

- Implement `kgram_wildcard_query(q)`
- Queries k-gram for matching words, given wildcard query
- Returns list of possible words

```
1 def kgram_wildcard_query(q):
2     '''
3     Query the k-gram index for words matching q. Return the matches as a set
4     '''
5     grams = wildcard_parse(q)
6     ### TODO for the assignment: execute the wildcard query on the k-gram index
7     kgram_matches = {}
8     post_filter = re.compile("^" + q.replace("*", "\\w*") + "$")
9     res = {r for r in kgram_matches if post_filter.match(r) is not None}
10    return res
```



Programming exercise

Spell checking

- Modify Jaccard threshold in first code snippet
- Implement `spellcorrect(word)`
- Compute the k-grams from the input word
- Find matching words for each k-gram
- For each candidate word, check Jaccard coefficient and filter

<https://create.kahoot.it/details/38402941-280b-43f9-94ba-4ea8805b9b16>

Exam questions

FS19 – Question 28

Compute the Jaccard coefficient for the following pairs of words, using their sets of 3-grams (with duplicates eliminated, if any). For example, the set of 3-grams contained in the word "time" is

{" \$ ti", "tim", "ime", "me \$" }

Each answer must be given as an irreducible fraction (or integer) .

Question: electric, electronic

Answer:

$g(\text{electric}) = \{\$el, ele, lec, ect, ctr, tri, ric, ic\$ \}$

$g(\text{electronic}) = \{\$el, ele, lec, ect, ctr, tro, ron, oni, nic, ic\$ \}$

Intersection: $\{\$el, ele, lec, ect, ctr, ic\$ \}$

$J(\text{electric}, \text{electronic}) = 6 / (8 + 10 - 6) = 0.5$

Exam questions

FS19 – Question 28

Compute the Jaccard coefficient for the following pairs of words, using their sets of 3-grams (with duplicates eliminated, if any). For example, the set of 3-grams contained in the word "time" is

{" \$ ti", "tim", "ime", "me \$" }

Each answer must be given as an irreducible fraction (or integer) .

Question: automobile, automation

Answer:

$g(\text{automobile}) = \{ \$au, aut, uto, tom, omo, mob, obi, bil, ile, le\$ \}$

$g(\text{automation}) = \{ \$au, aut, uto, tom, oma, mat, ati, tio, ion, on\$ \}$

Intersection: $\{ \$au, aut, uto, tom \}$

$J(\text{automobile}, \text{automation}) = 4 / (10 + 10 - 4) = 0.25$

Exam questions

FS19 – Question 28

Compute the Jaccard coefficient for the following pairs of words, using their sets of 3-grams (with duplicates eliminated, if any). For example, the set of 3-grams contained in the word "time" is

{ "\$ti", "tim", "ime", "me\$" }

Each answer must be given as an irreducible fraction (or integer).

Question: information, retrieval

Answer:

0

Exam questions

FS19 – Question 28

Compute the Jaccard coefficient for the following pairs of words, using their sets of 3-grams (with duplicates eliminated, if any). For example, the set of 3-grams contained in the word "time" is

{ "\$ti", "tim", "ime", "me\$" }

Each answer must be given as an irreducible fraction (or integer).

Question: fifteen,fifteen

Answer:

1

Exam questions

FS19 – Question 32

How many rotations does the following single term contribute to a permuterm index?

mama

\$mama, a\$mam, ma\$ma, ama\$m, mama\$

FS21 – Question 21

For each one of the following terms, state whether or not it falsely matches the wildcard query «some*e» if the evaluation of the query is done by simply using a conjunction of 3-grams, and with no post-processing.

sometime

No, since it correctly matches the wildcard query.

FS21 – Question 21

For each one of the following terms, state whether or not it falsely matches the wildcard query «some*e» if the evaluation of the query is done by simply using a conjunction of 3-grams, and with no post-processing.

some

Yes, since it is a false positive. (* is missing)

FS21 – Question 21

For each one of the following terms, state whether or not it falsely matches the wildcard query «some*e» if the evaluation of the query is done by simply using a conjunction of 3-grams, and with no post-processing.

sommelier

No, since the 3-gram «ome» doesn't occur in the term it will not be returned.

FS21 – Question 21

For each one of the following terms, state whether or not it falsely matches the wildcard query «some*e» if the evaluation of the query is done by simply using a conjunction of 3-grams, and with no post-processing.

something

Yes. «something» covers all 3-grams of the wildcard query, so it is a false positive.