Dumpy: A Compact and Adaptive Index for Large Data Series Collections

Zeyu Wang
Fudan Univerisity
Shanghai, China

Qitong Wang
Université Paris Cité
Paris, France

Peng Wang
Fudan University
Shanghai, China

Themis Palpanas
Université Paris Cité &
French University Institute (IUF)
Paris, France

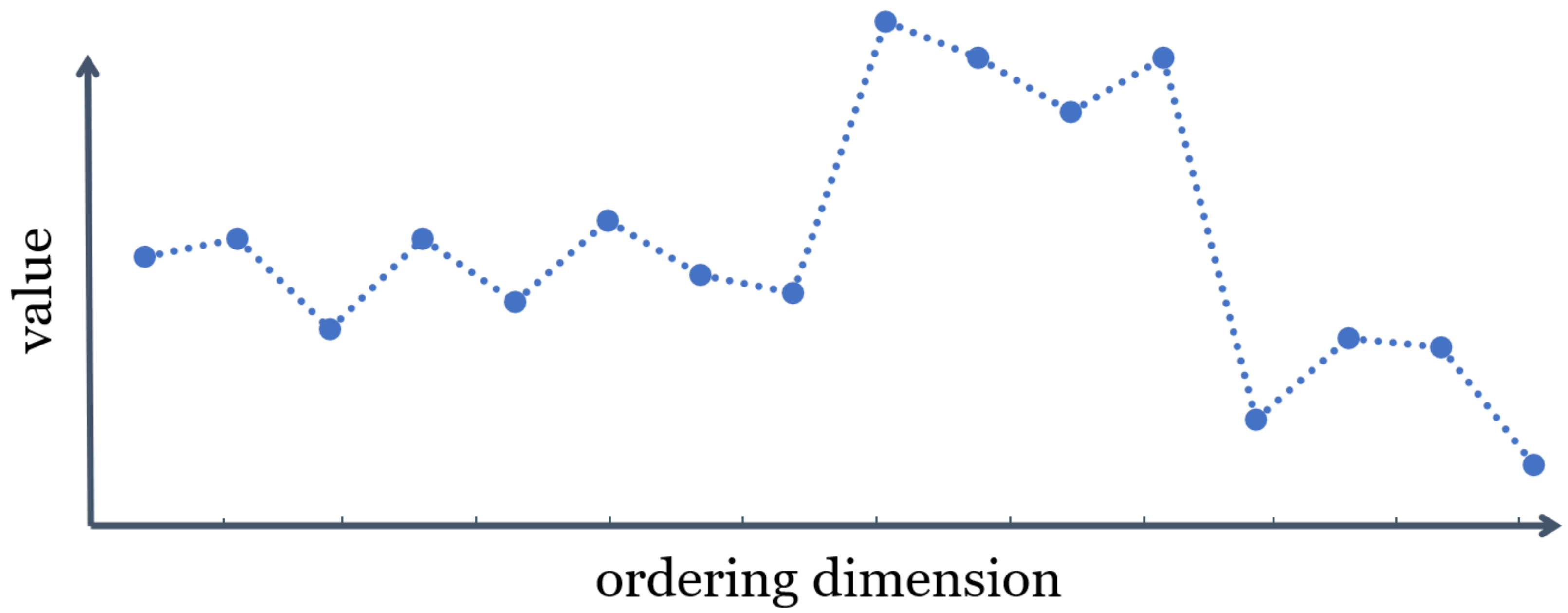
Wei Wang
Fudan University
Shanghai, China

ACM SIGMOD 2023, Seattle

paper: https://helios2.mi.parisdescartes.fr/~themisp/publications/sigmod23-dumpy.pdf
video: https://files.atypon.com/acm/99f6febc21ad6c5a979f504caf188d9a
code: https://github.com/DSM-fudan/Dumpy

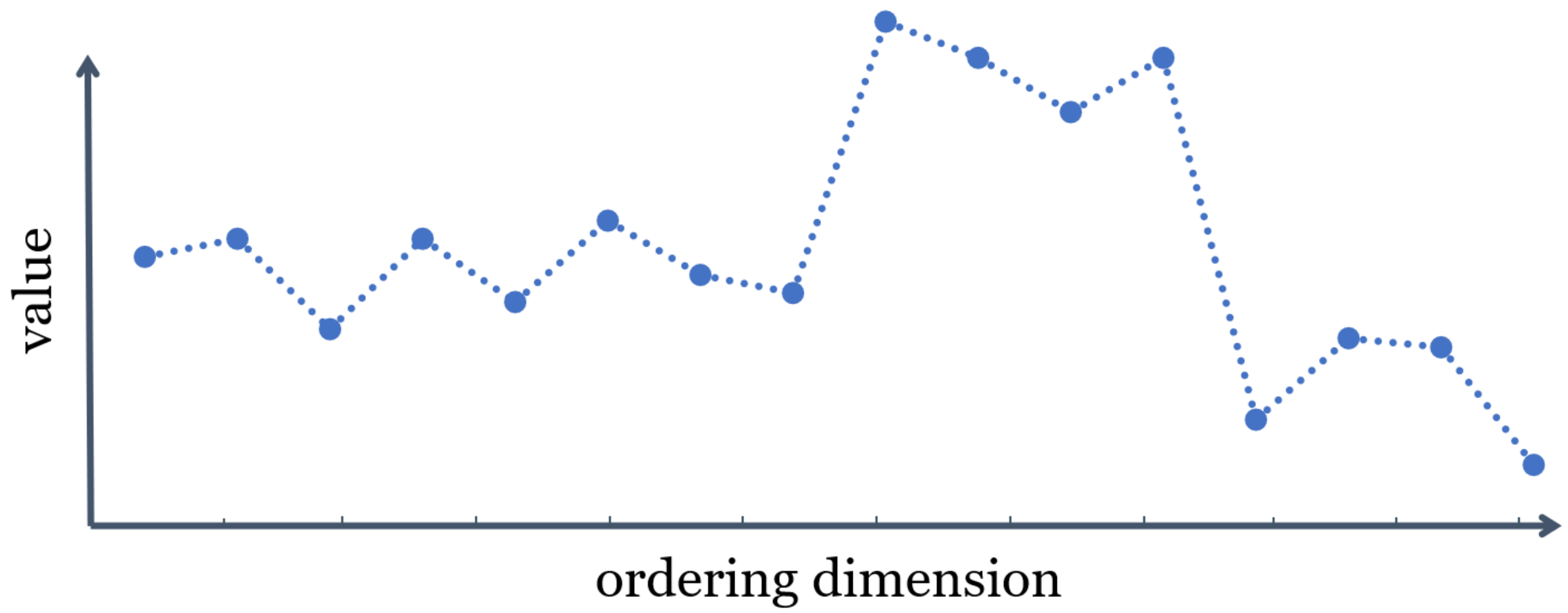
1-Data Series

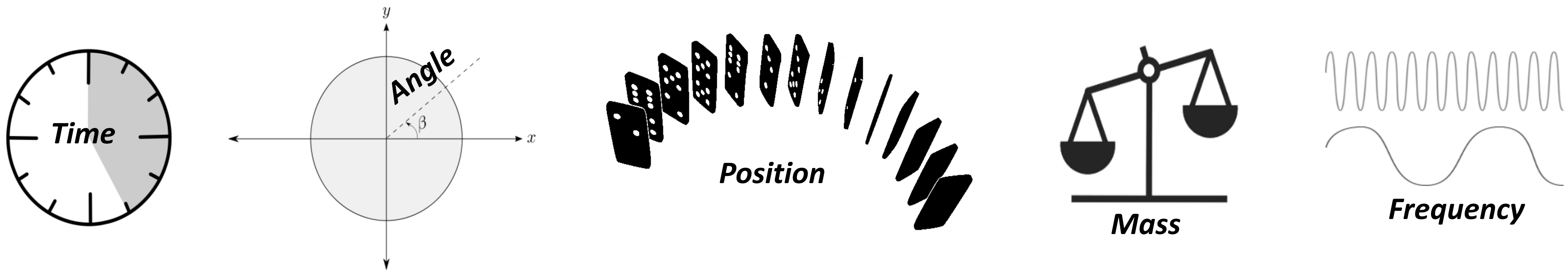
- Sequence of points ordered along some dimension



1-Data Series

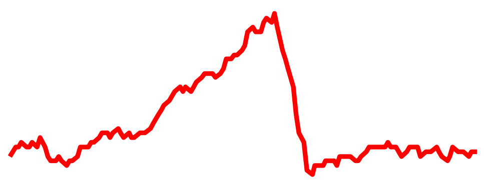
- Sequence of points ordered along some dimension





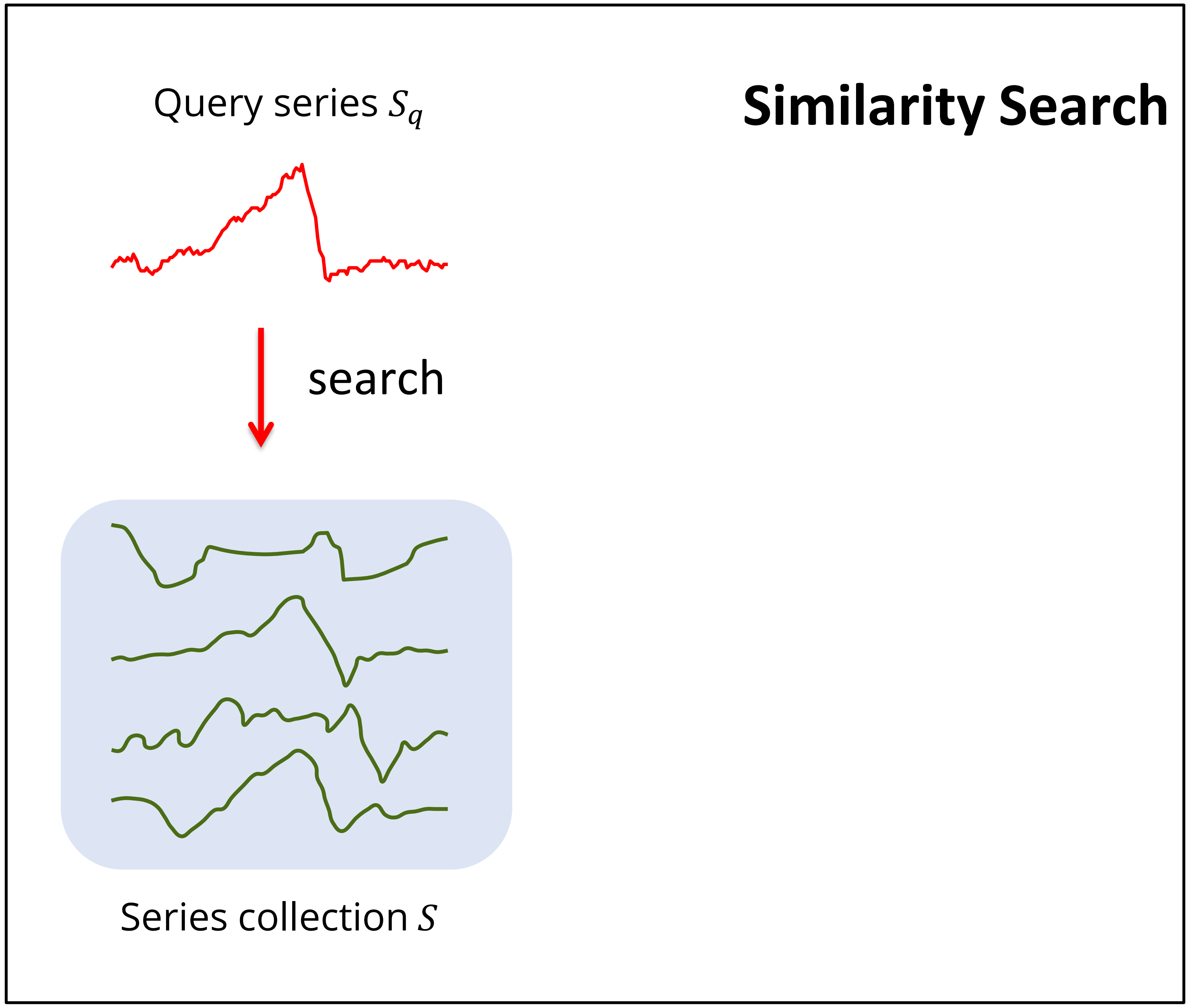
1-Data Series Similarity Search

Query series S_q

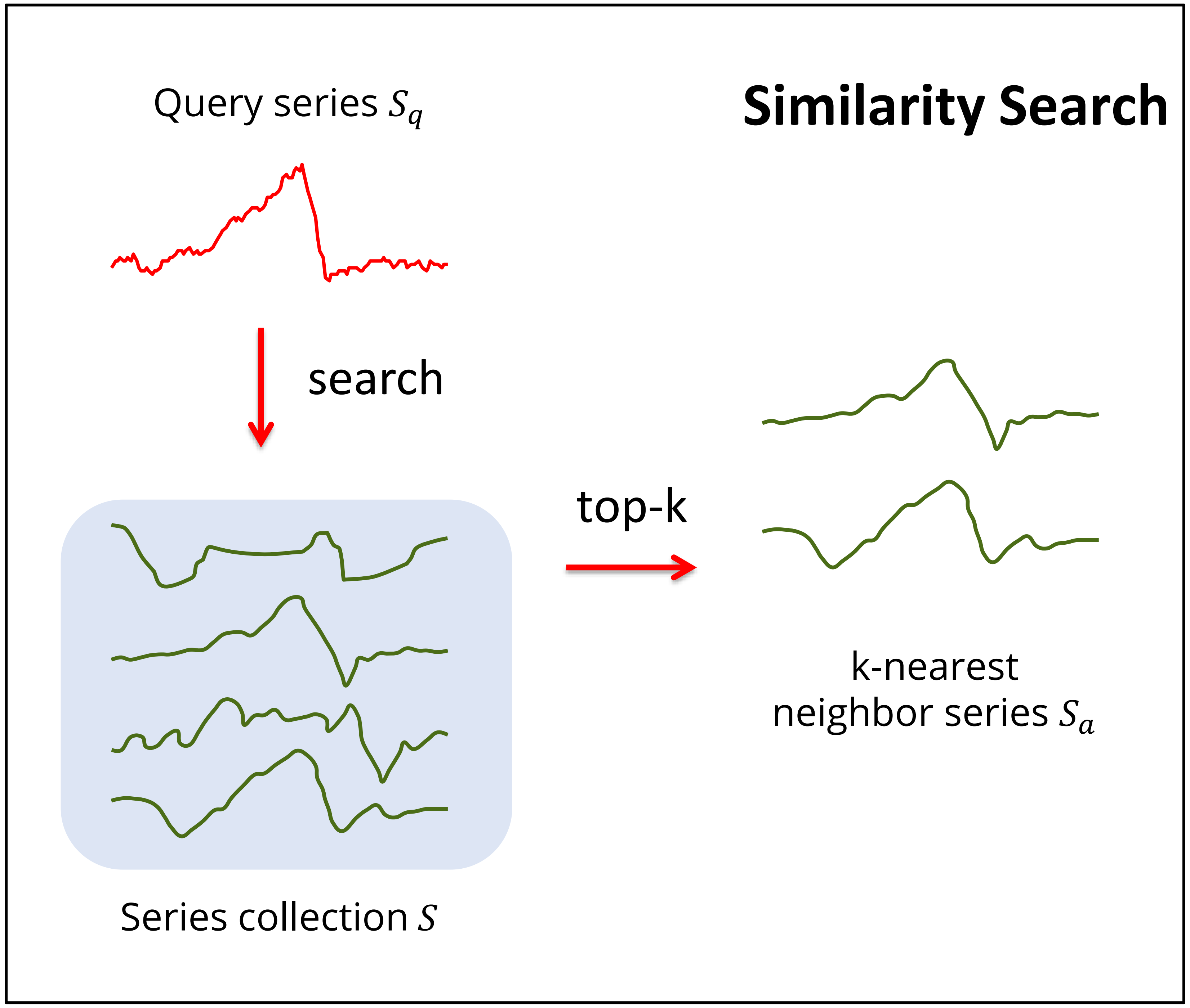


Similarity Search

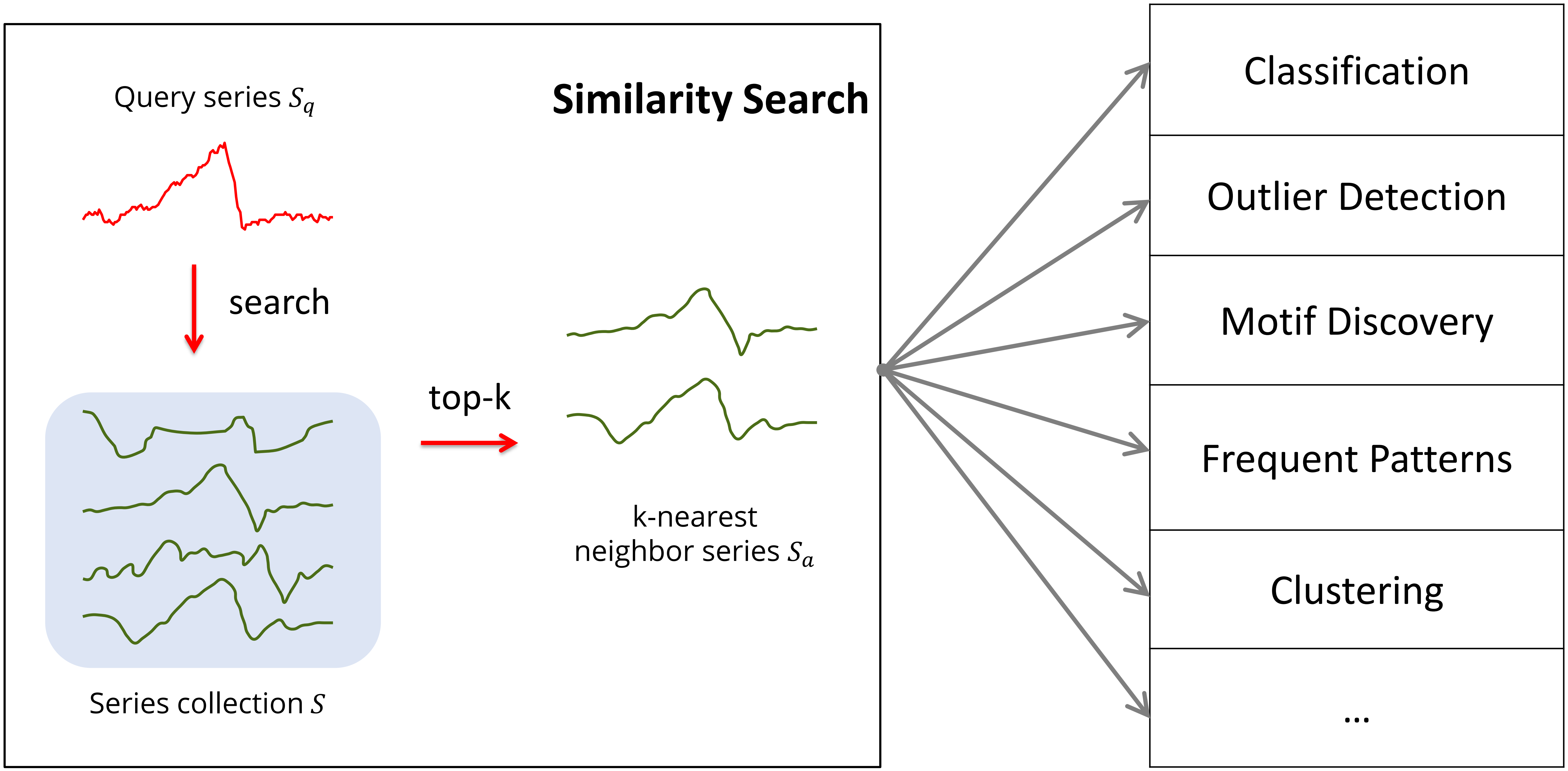
1-Data Series Similarity Search



1-Data Series Similarity Search



1-Data Series Similarity Search



1-Problem Definition

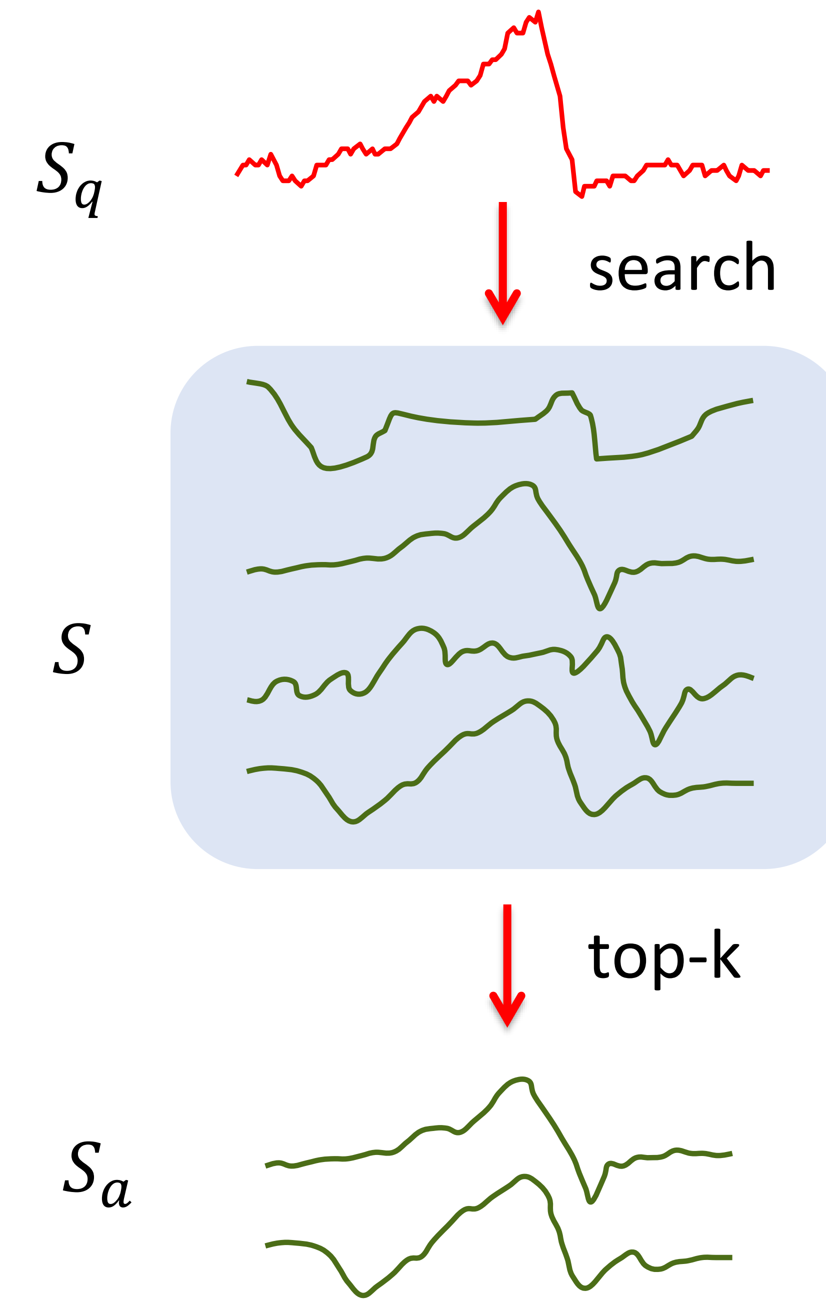
- Similarity search
  - given a series set S, a query series S_q and a similarity measure $d(\cdot,\cdot)$
    - d is commonly the Euclidean distance (ED) or Dynamic Time Warping (DTW)
  - find the closest series in S to s_q, i.e.,

$$s_a = \arg\min_{s_i \in S} d(s_q, s_i)$$

Exact answers sometimes unnecessary

- Approximate similarity search
  - find $s_{a'}$,$d(s_q, s_{a'}) \approx d(s_q, s_a)$
  - core requirements: Accuracy & Scalability

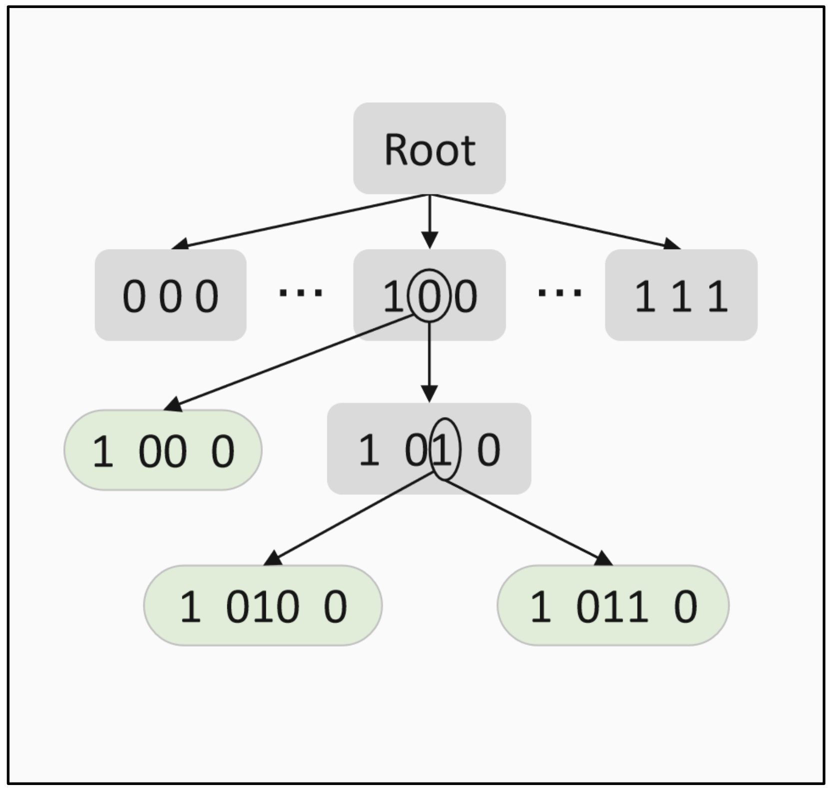
Data Series Index



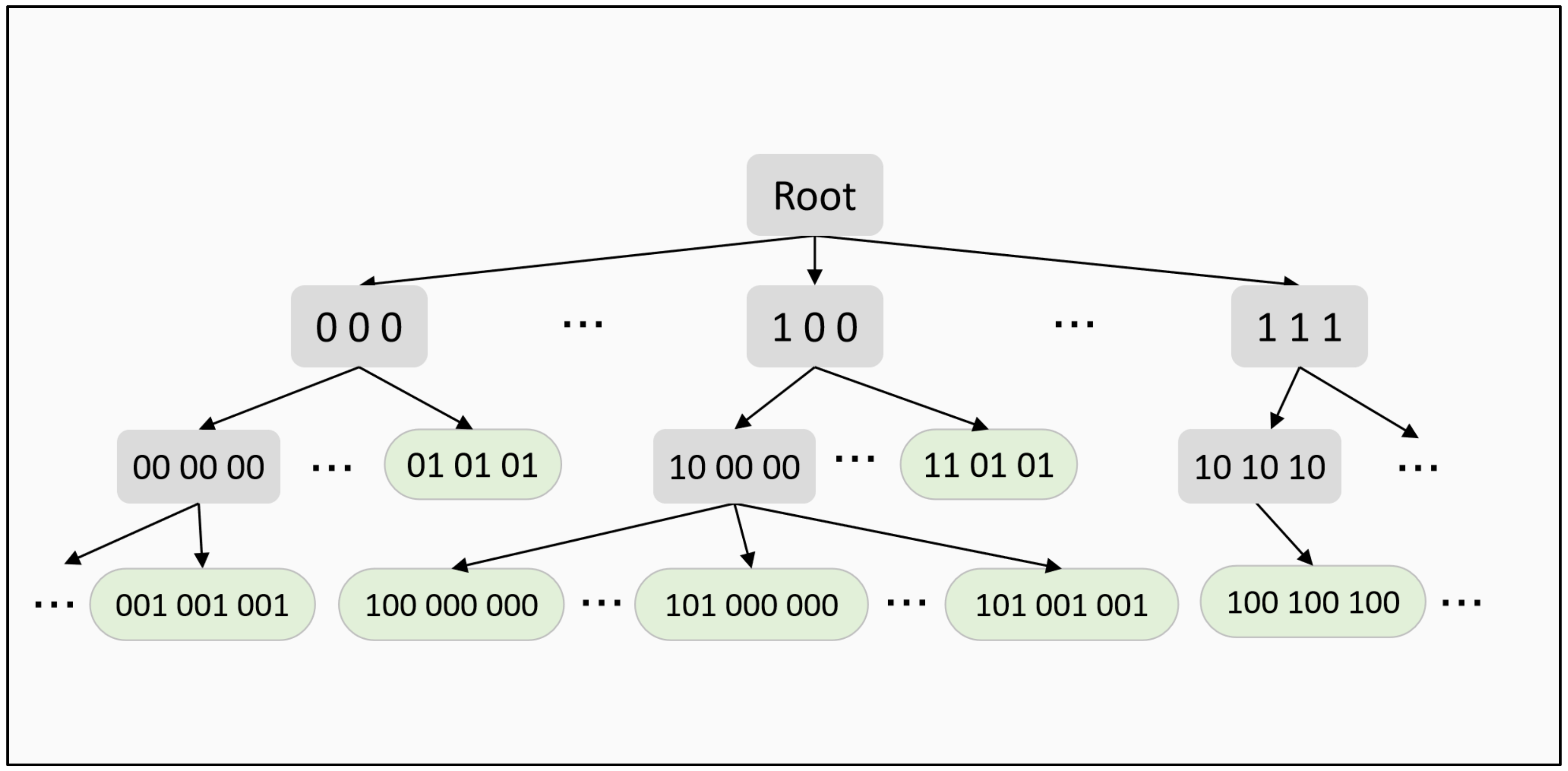
2-Limitations of Existing Solutions

Two structures of iSAX-based index

2-ary[1]



Full-Ary[2]



Splitting Decisions

Not Accurate ↔ **Not Scalable**

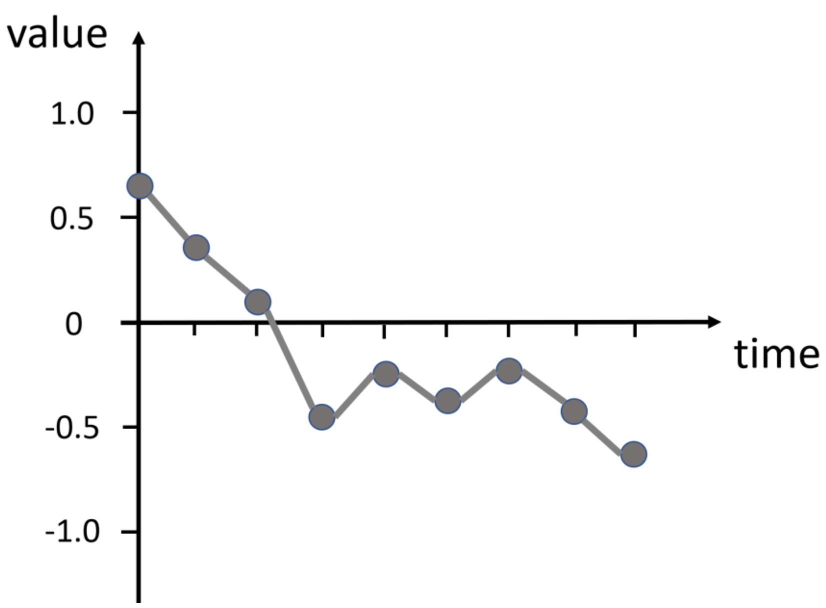
Problem 1: What's the right splitting decision between these two extremes?

Problem 2: How to efficiently implement splitting?

1. Alessandro Camerra, et al. Beyond One Billion Time Series: Indexing and Mining Very Large Time Series Collections with iSAX2+. KAIS 39(1):123-151, 2014.
2. Zhang, L. et al. 2019. TARDIS: Distributed Indexing Framework for Big Time Series Data. 2019 IEEE 35th International Conference on Data Engineering (ICDE)

3-iSAX Index Family

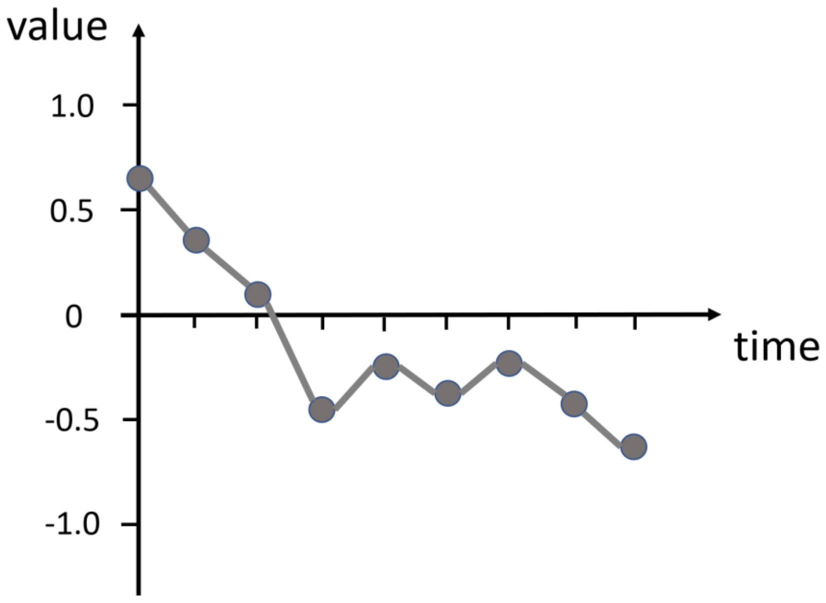
Raw series



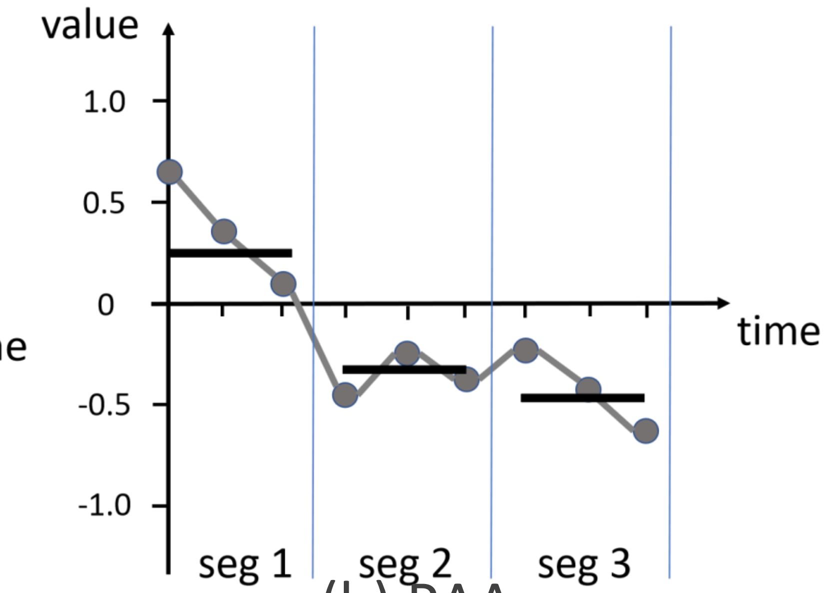
(a) raw series s

3-iSAX Index Family

Raw series → PAA approximation



(a) raw series s

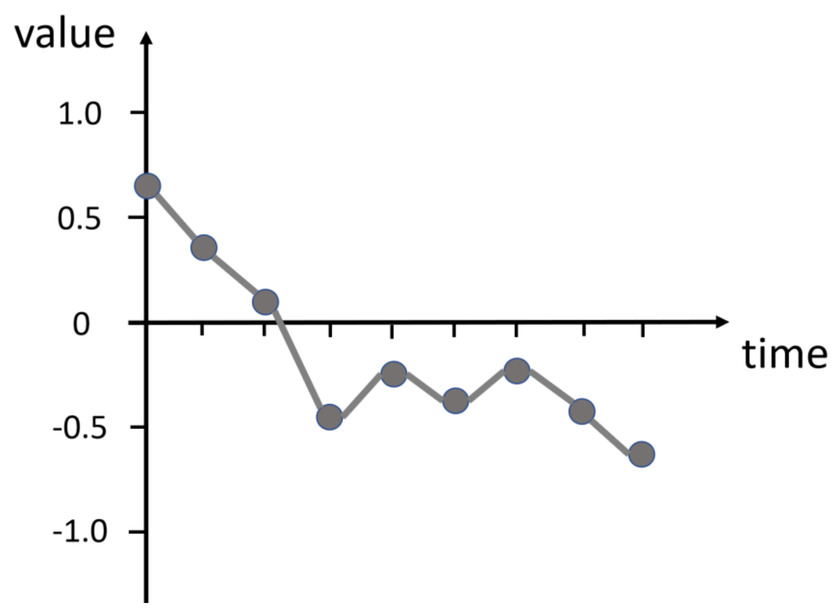


(b) PAA summarization

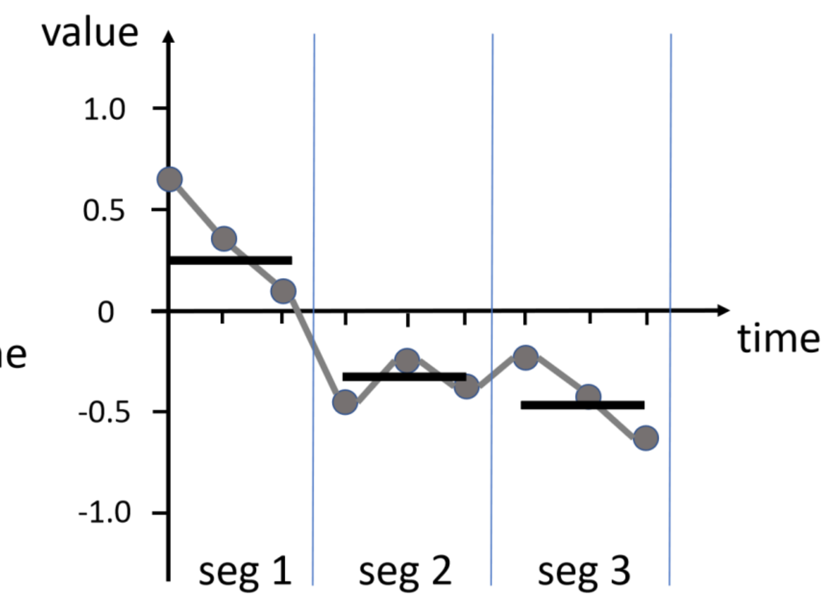
PAA(s)=[0.25, -0.3, -0.55]

3-iSAX Index Family

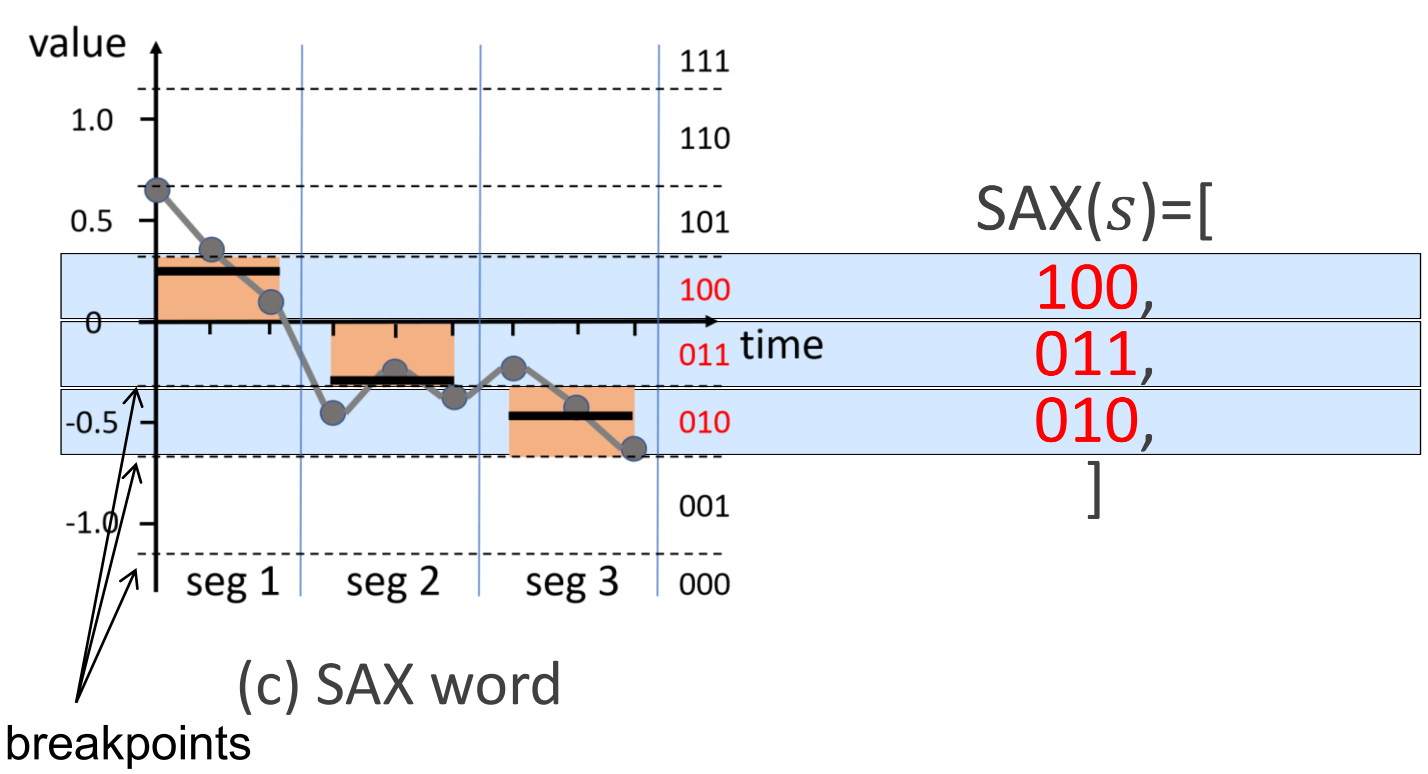
Raw series → PAA approximation → SAX symbolization



(a) raw series



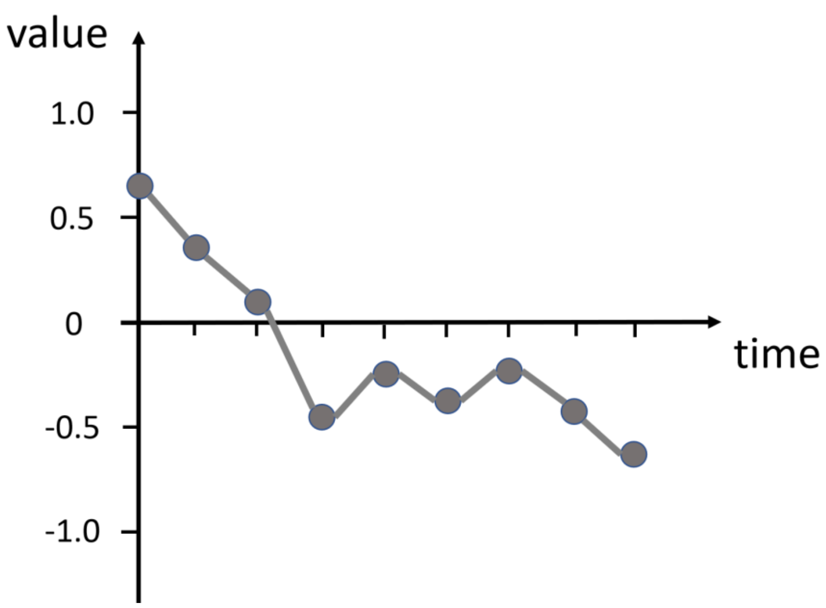
(b) PAA summarization



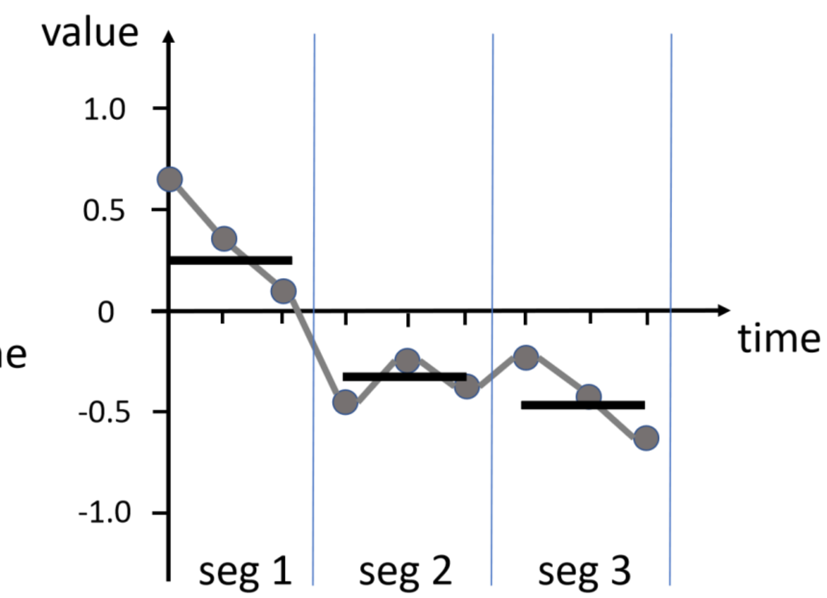
(c) SAX word

3-iSAX Index Family

Raw series → PAA approximation → SAX symbolization → iSAX index

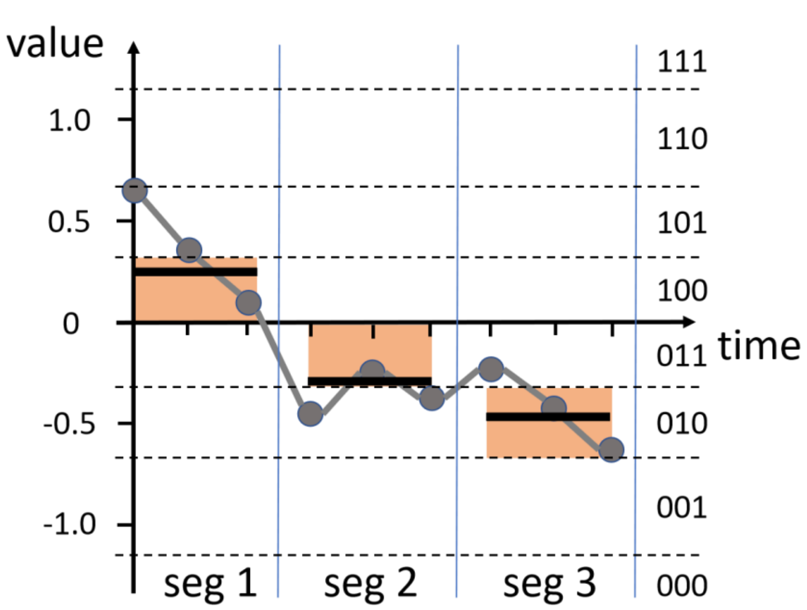


(a) raw series

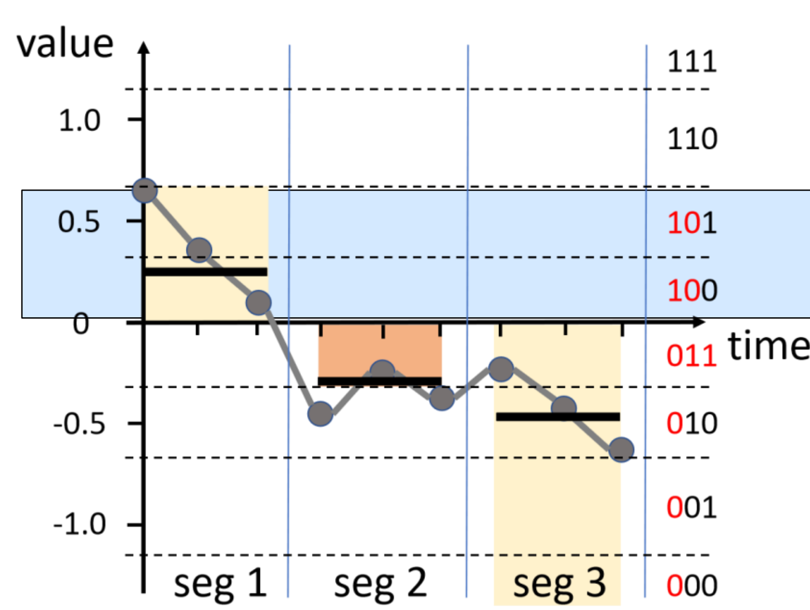


(b) PAA summarization

SAX(s)=[100, 011, 010]



(c) SAX word



(d) iSAX word

One iSAX word of s

iSAX(s)=[

10,

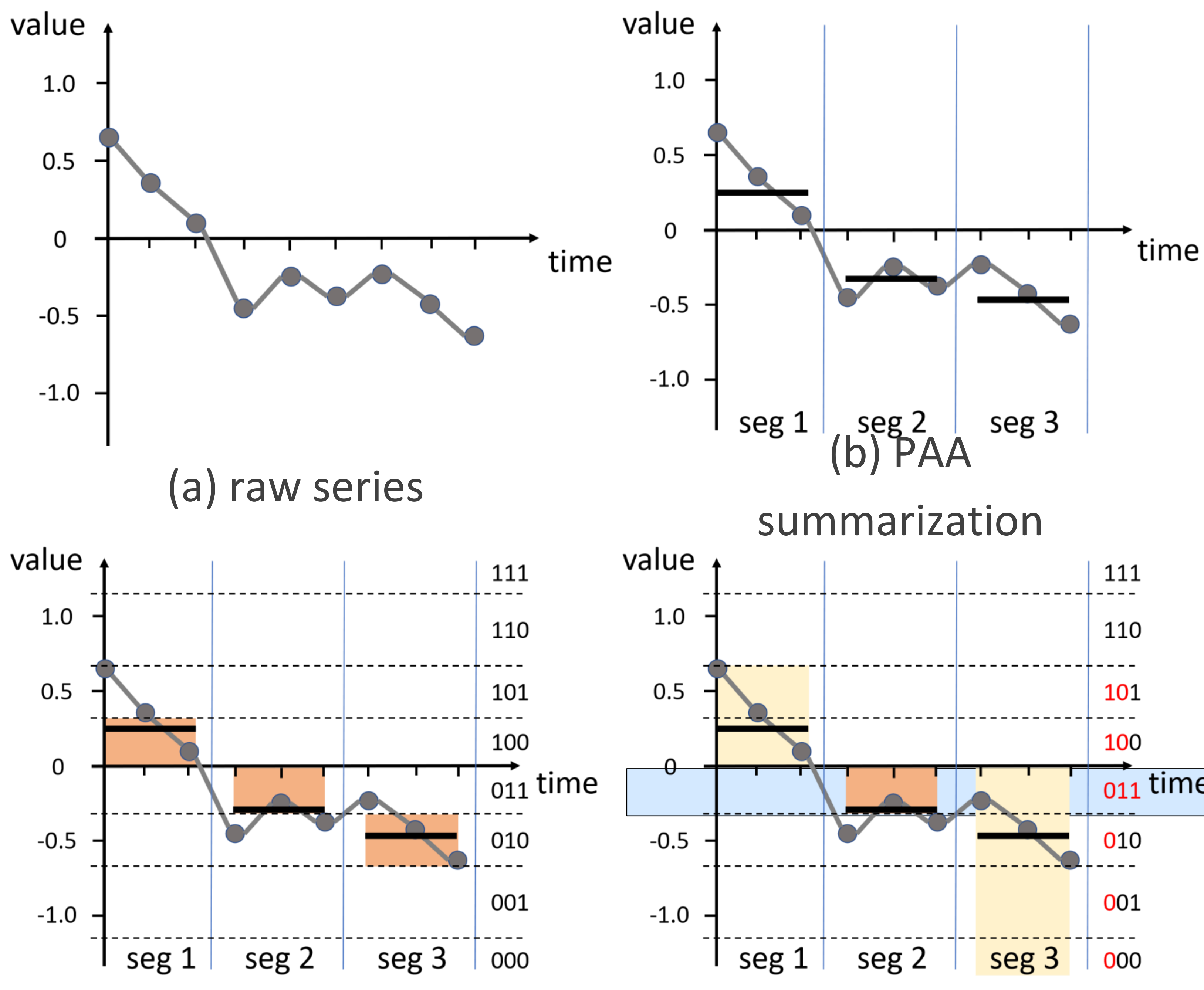
011,

0,

]

3-iSAX Index Family

Raw series → PAA approximation → SAX symbolization → iSAX index



(a) raw series

(b) PAA summarization

(c) SAX word

(d) iSAX word

SAX(s)=[100, 011, 010]

One iSAX word of s

iSAX(s)=[

10,

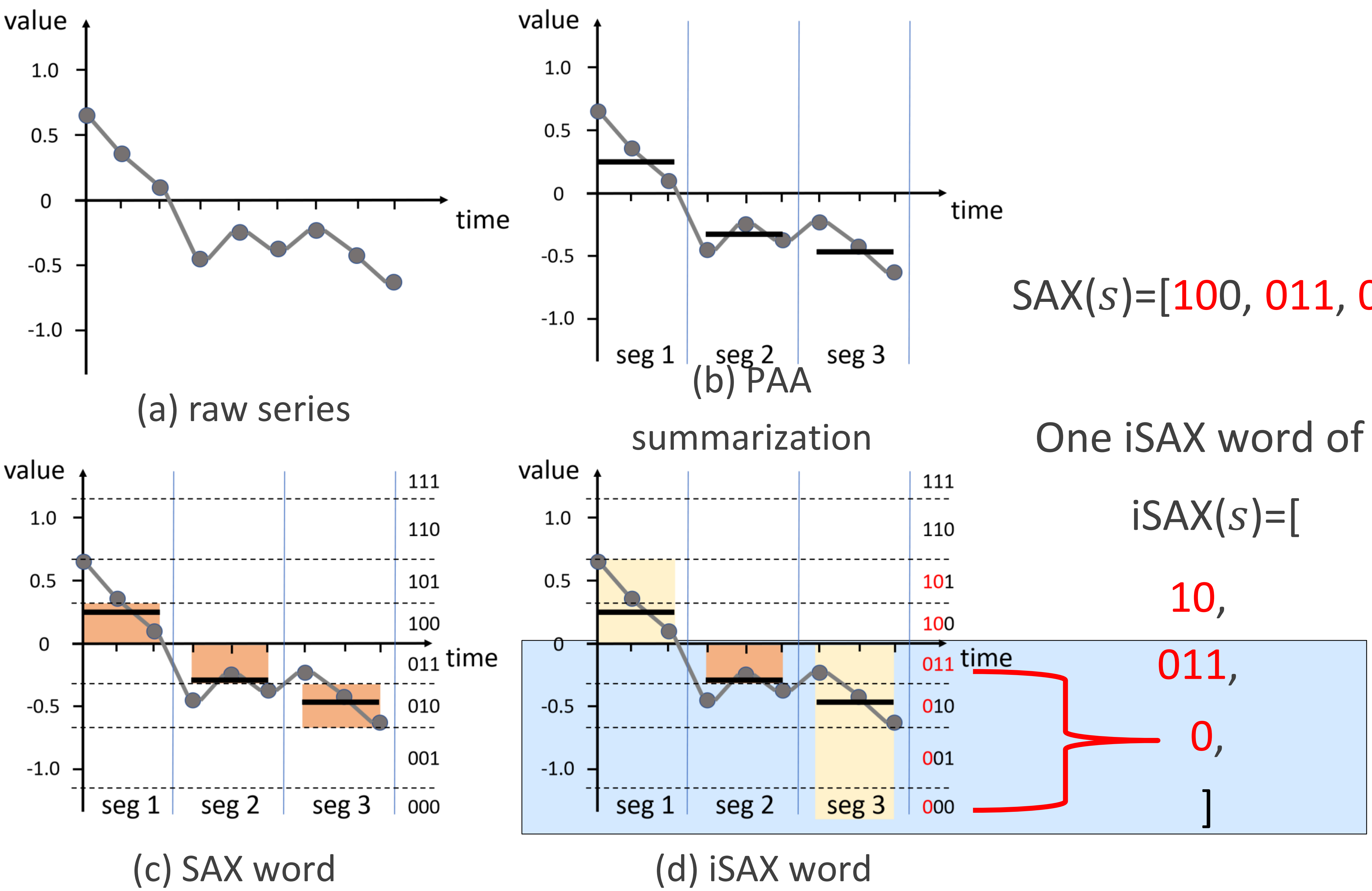
011,

0,

]

3-iSAX Index Family

Raw series → PAA approximation → SAX symbolization → iSAX index



SAX(s)=[100, 011, 010]

One iSAX word of s

iSAX(s)=[

10,

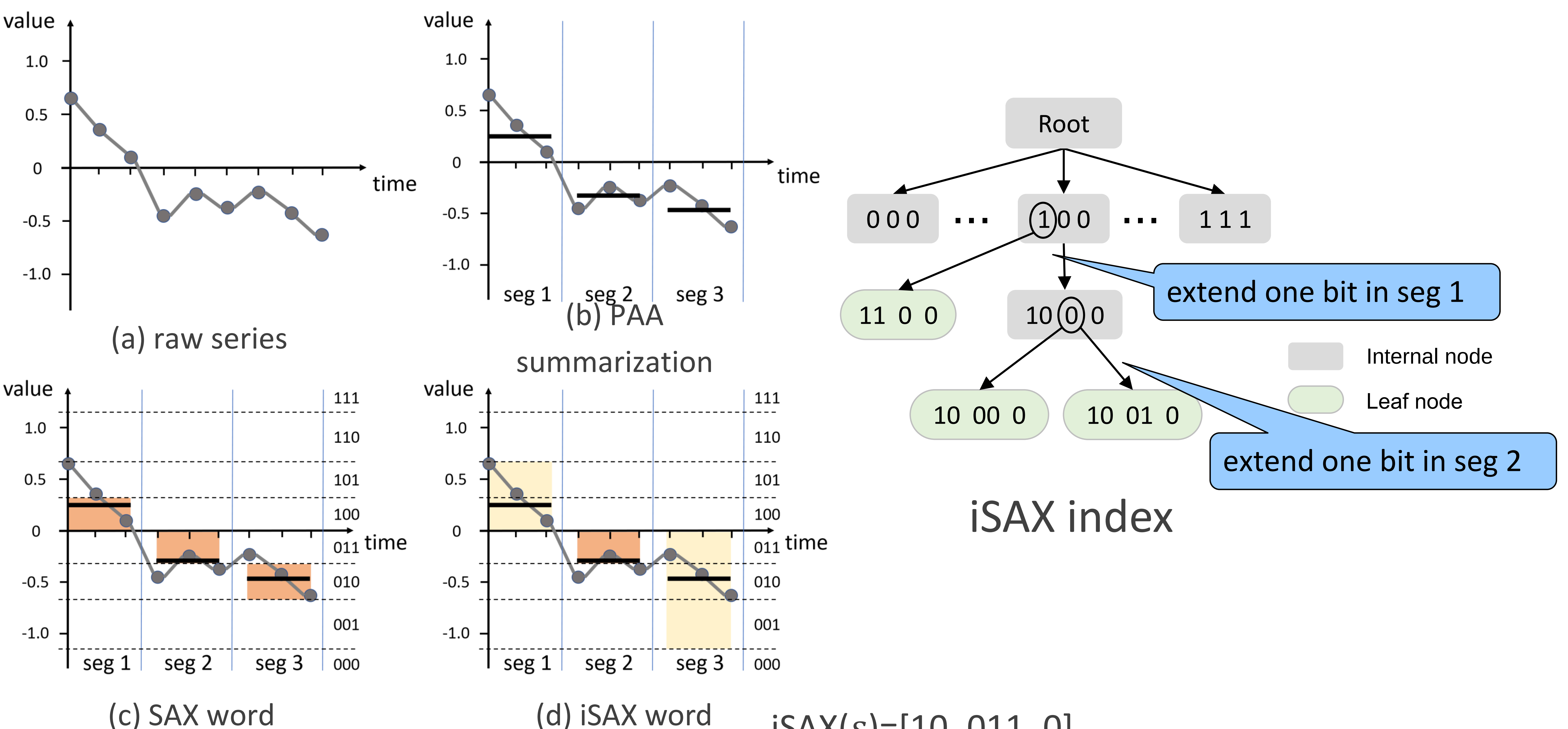
011,

0,

]

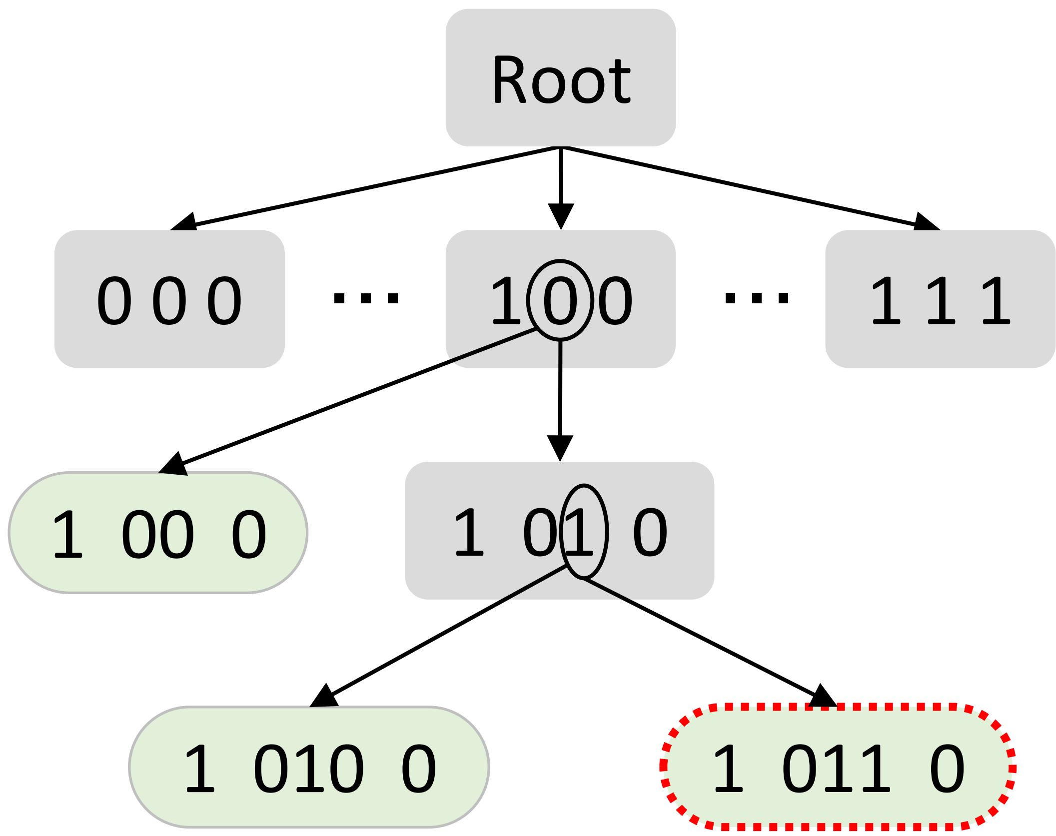
3-iSAX Index Family

Raw series → PAA approximation → SAX symbolization → iSAX index

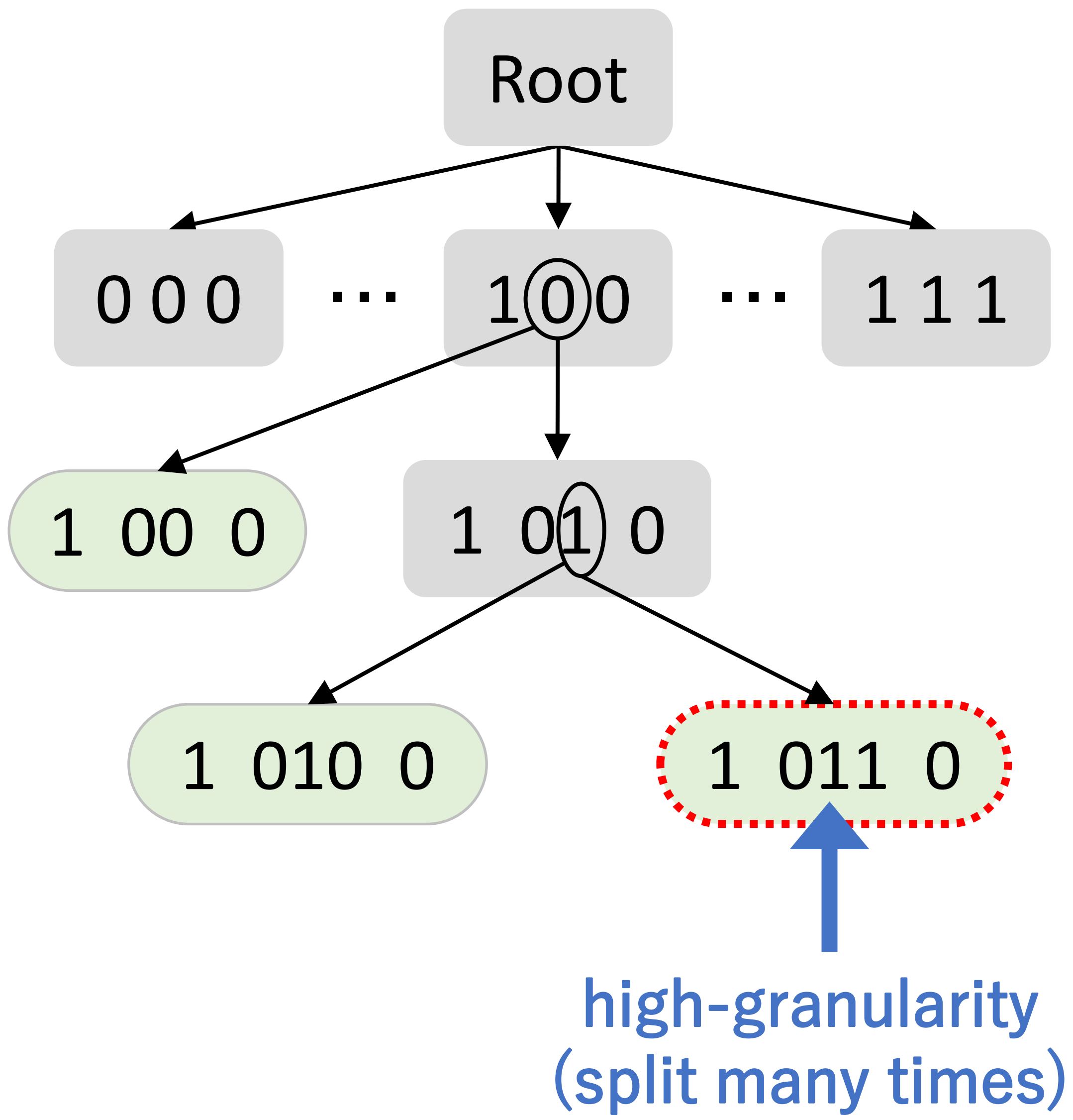


iSAX(s)=[10, 011, 0]

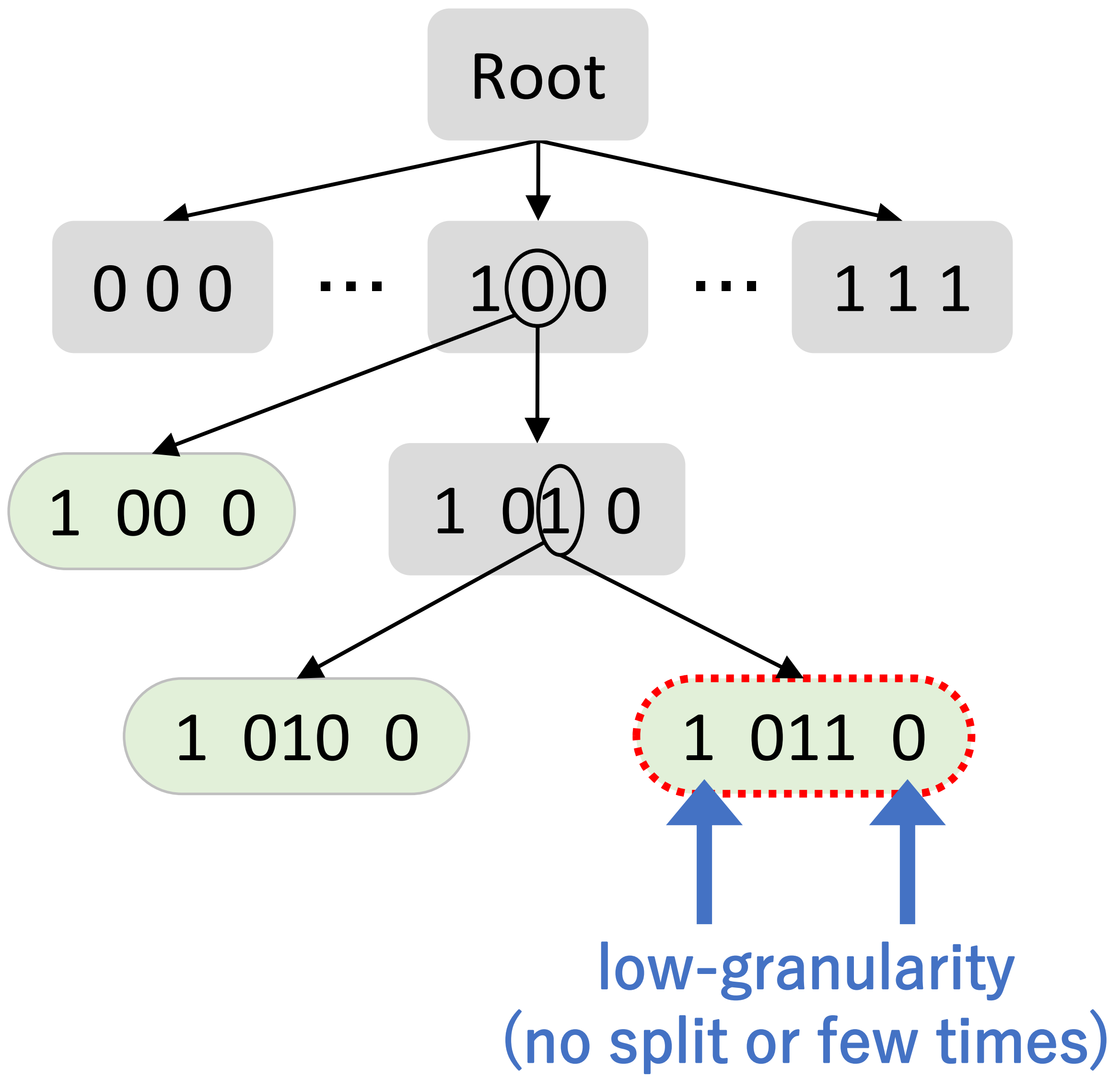
4-Proximity Problem of 2-ary iSAX Index



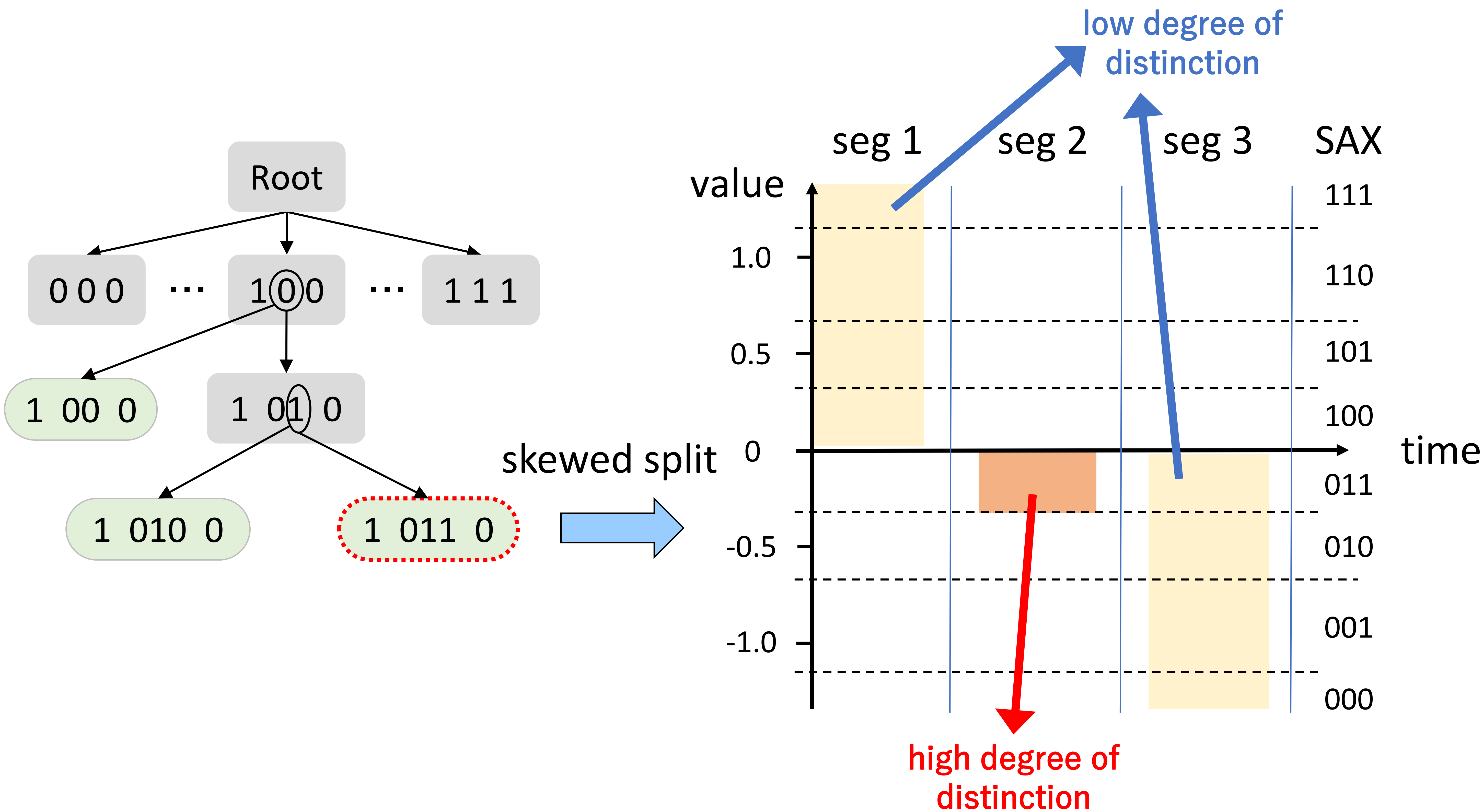
4-Proximity Problem of 2-ary iSAX Index



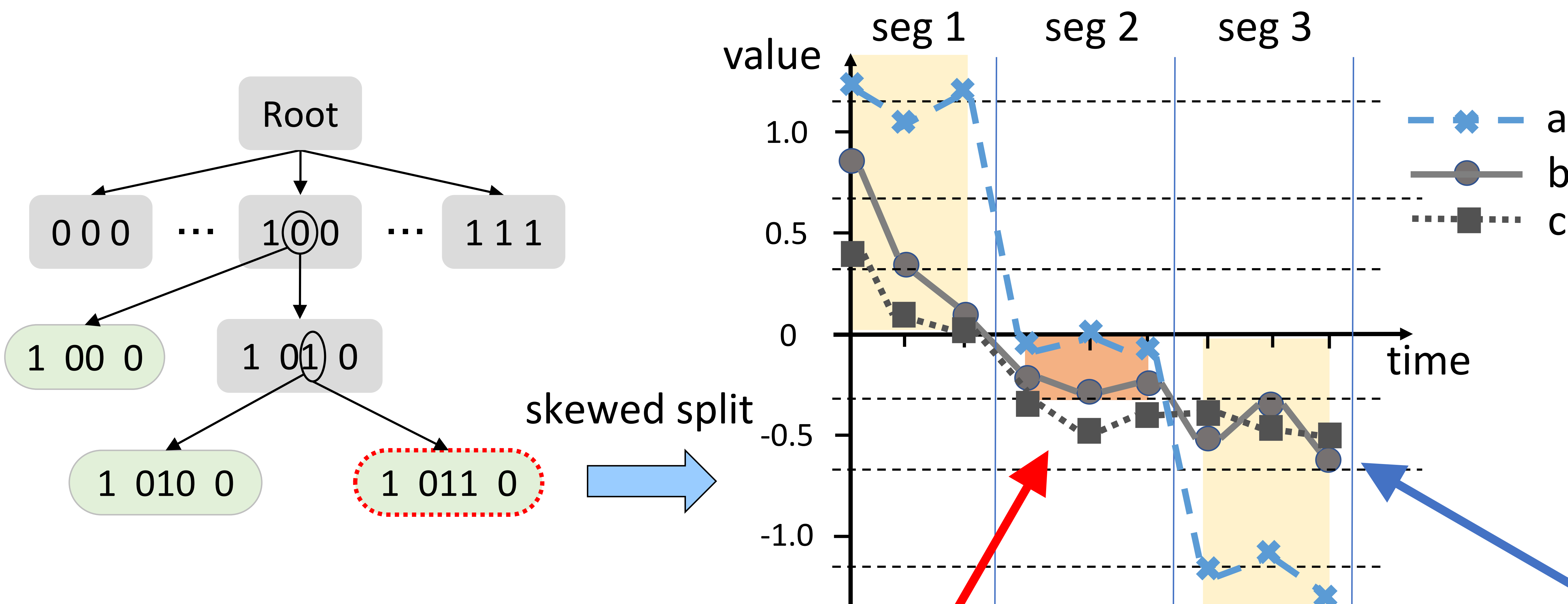
4-Proximity Problem of 2-ary iSAX Index



4-Proximity Problem of 2-ary iSAX Index



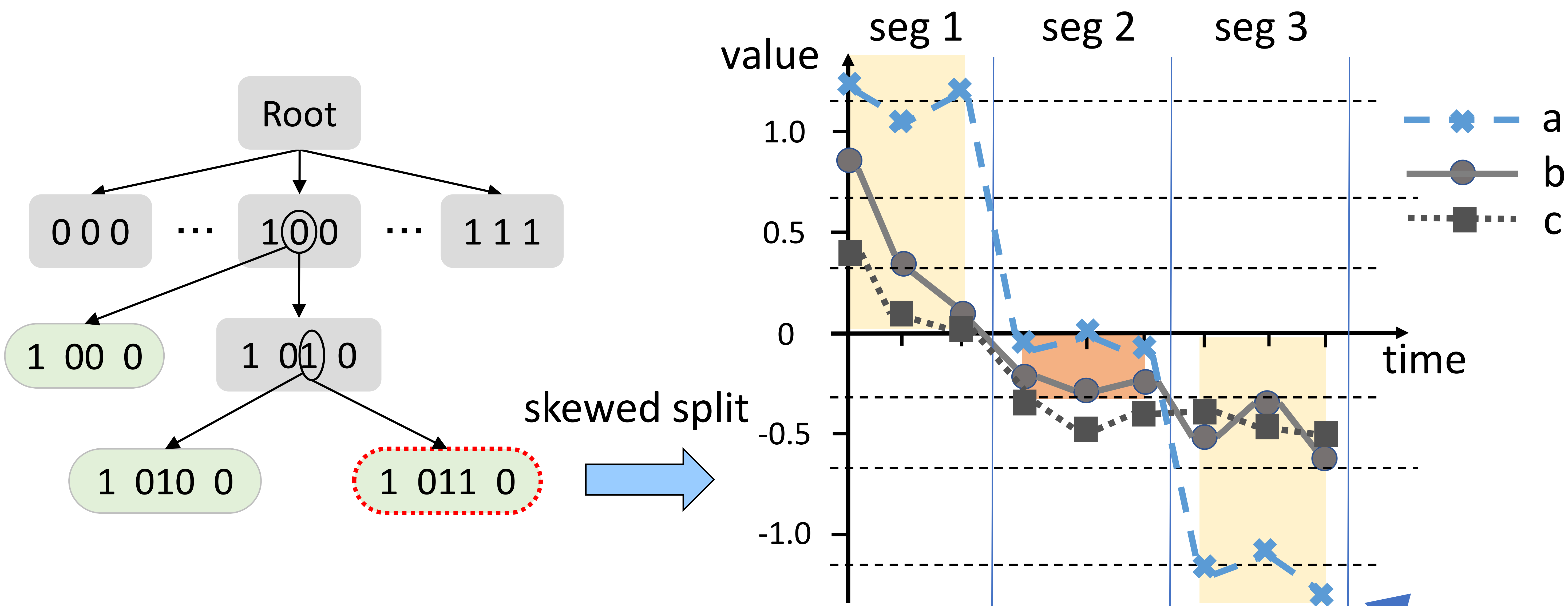
4-Proximity Problem of 2-ary iSAX Index



c excluded in this node for seg 2

b and c are very close

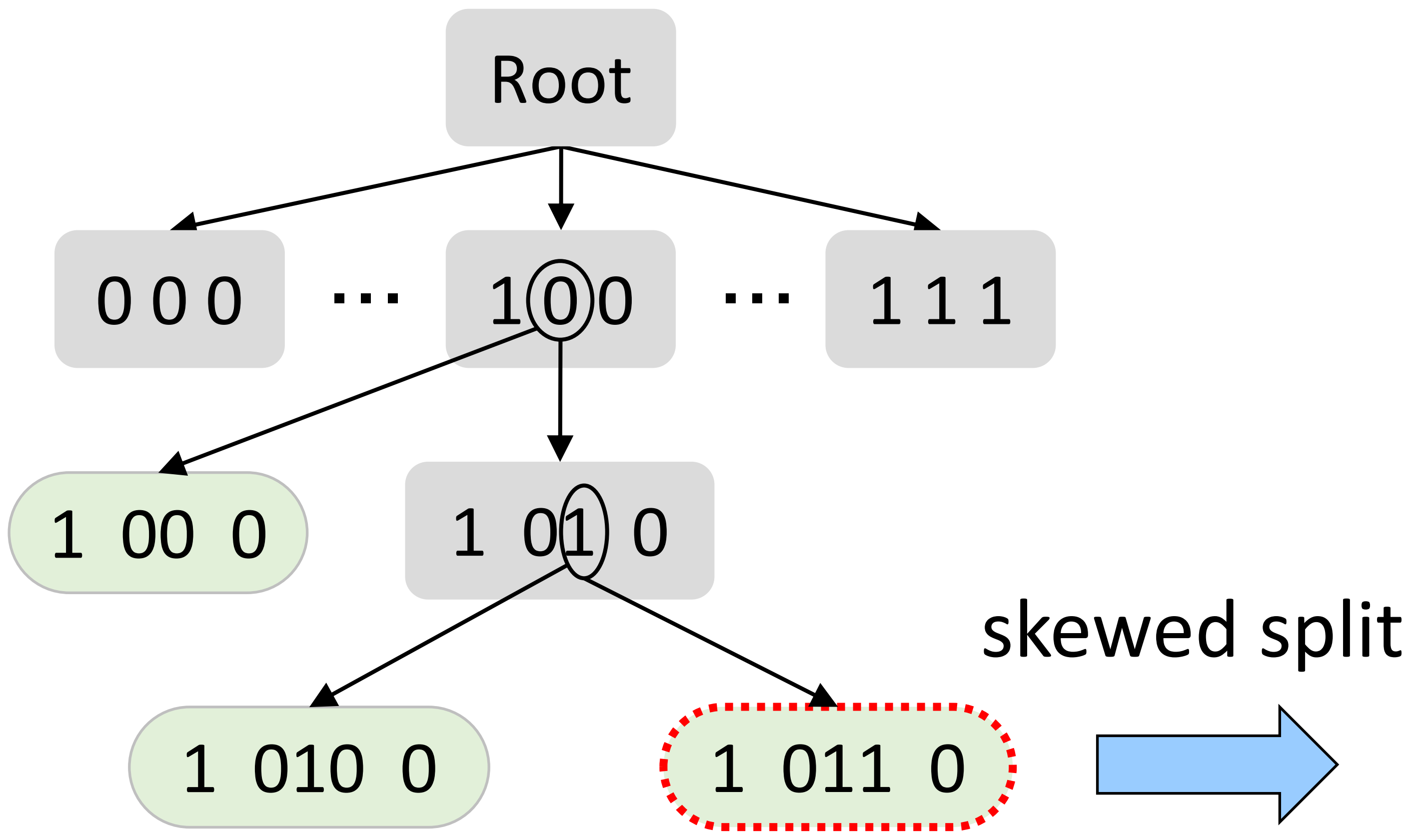
4-Proximity Problem of 2-ary iSAX Index

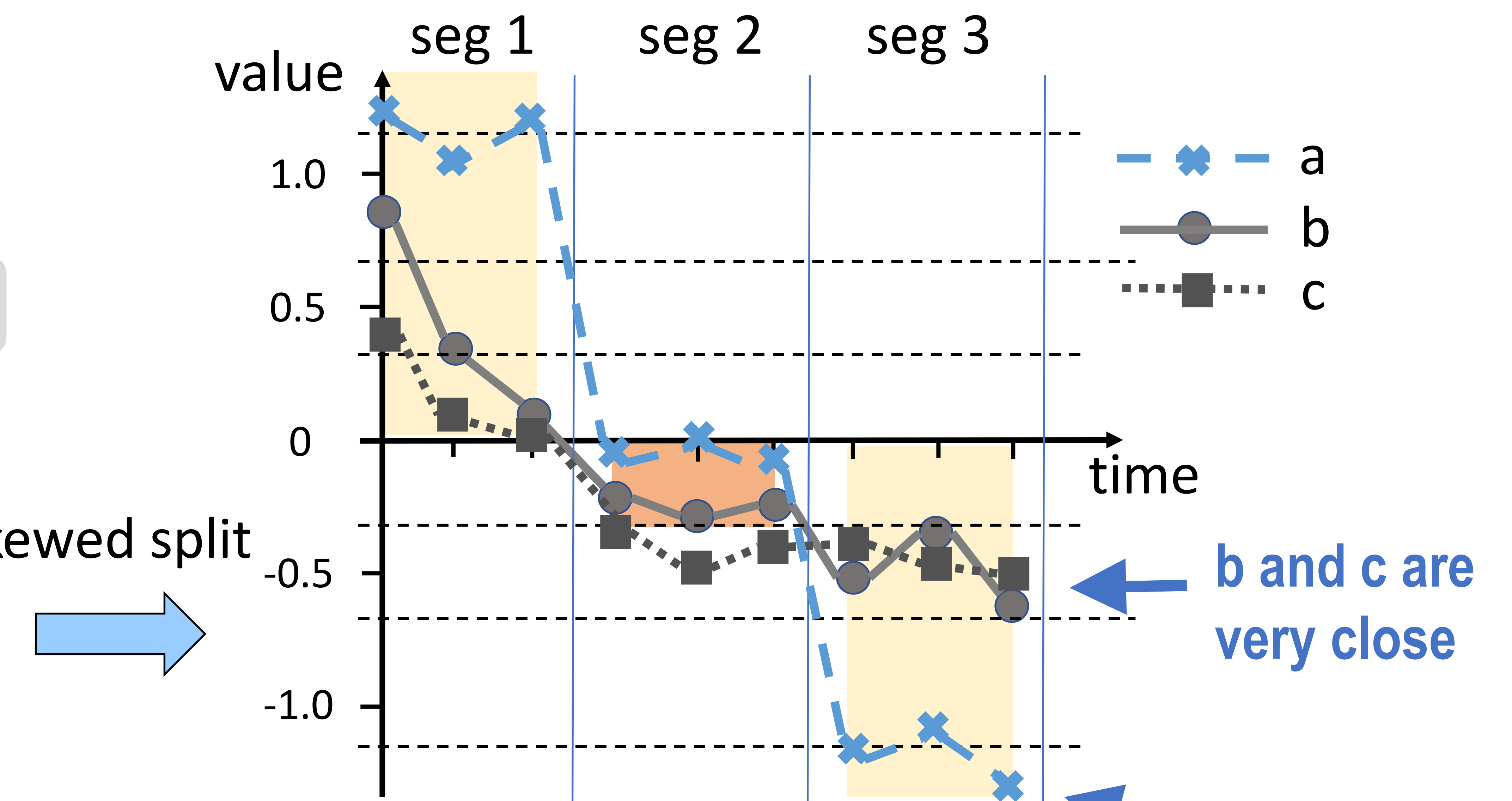


a is far from b and c

a is included in this node together with b

4-Proximity Problem of 2-ary iSAX Index

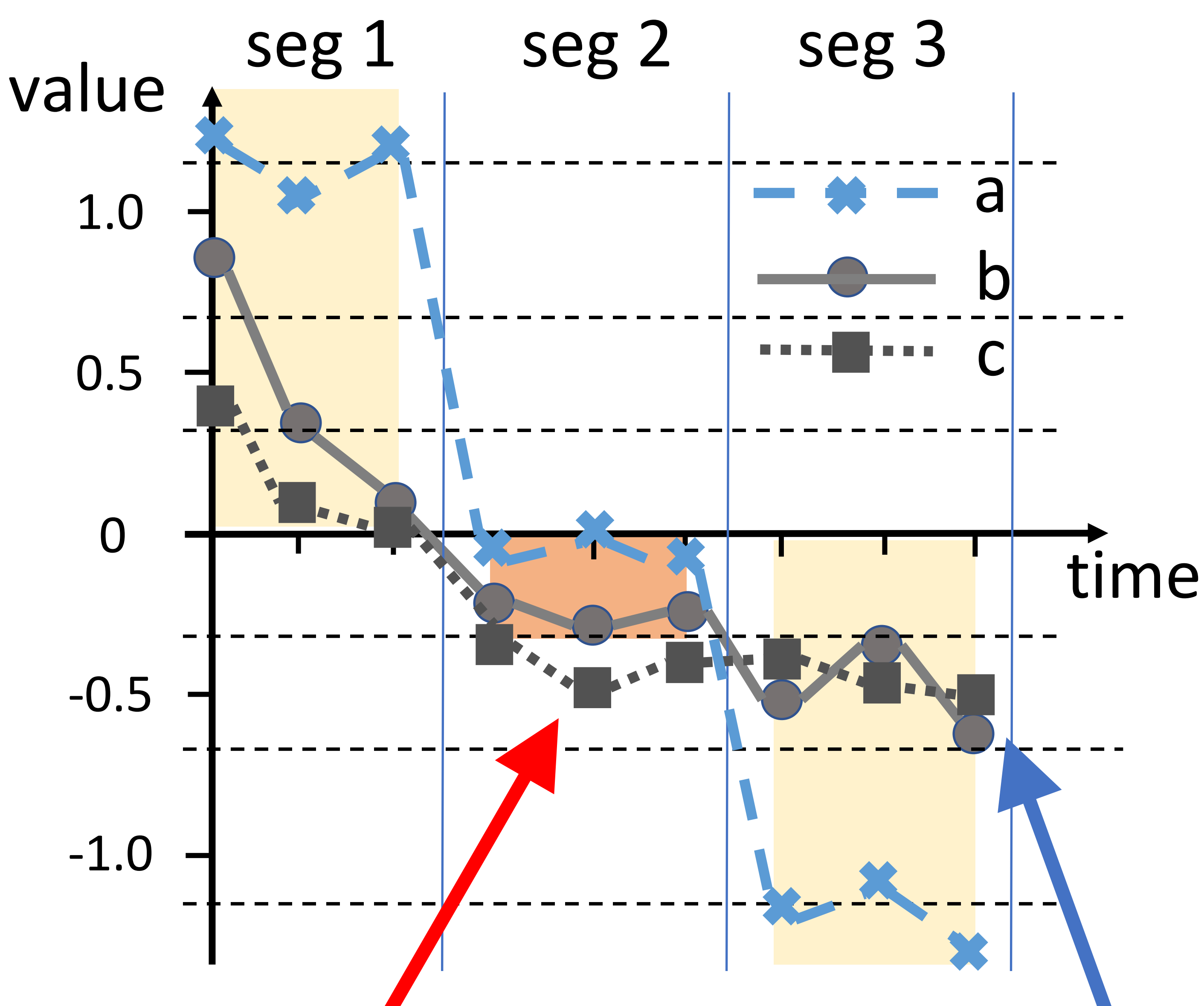




Reason:
<u>over-focus</u> on similarity on seg 2, but final distance equally contributed by all 3 segments

4-Proximity Problem of 2-ary iSAX Index

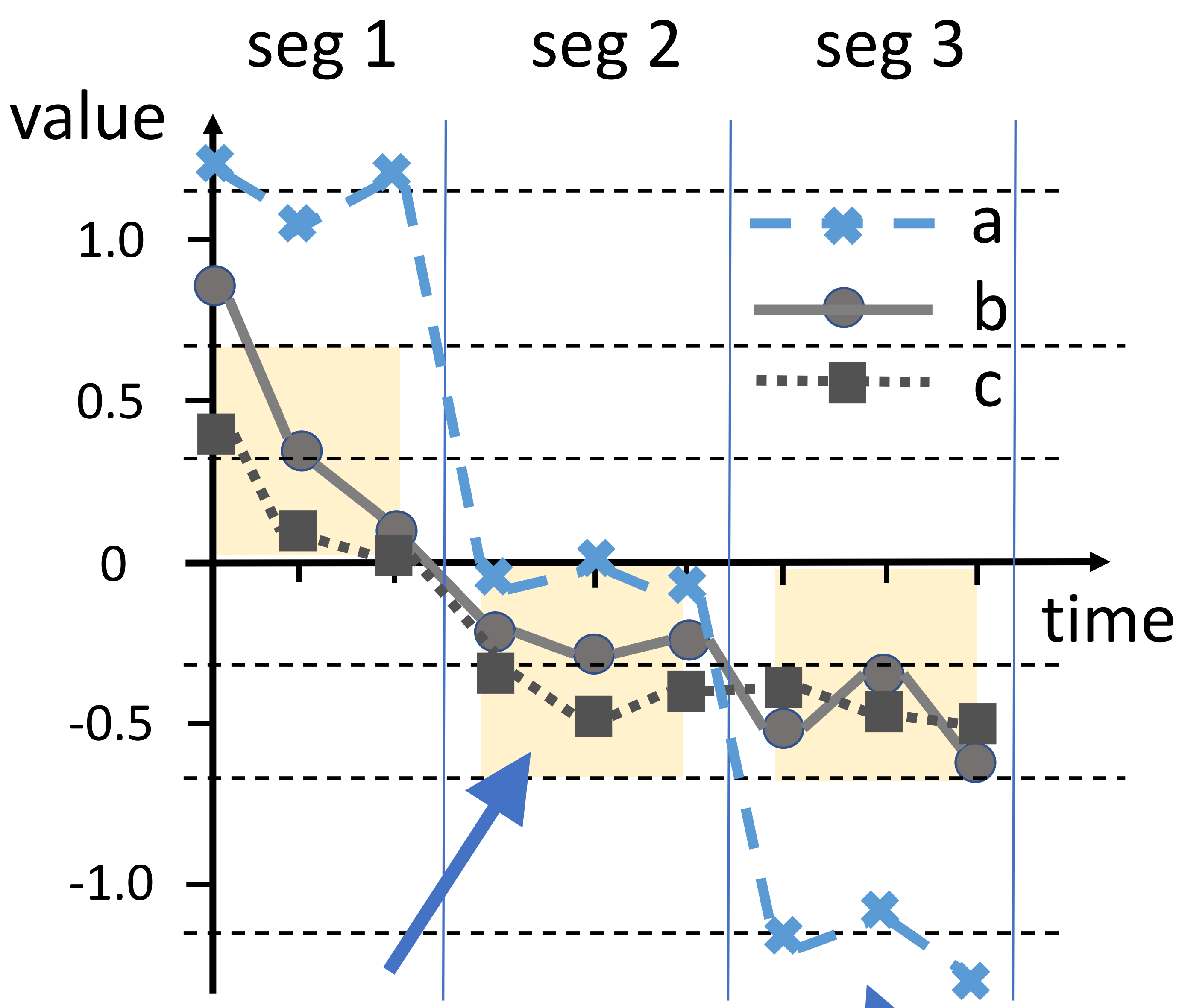
Skewed Split



c excluded in this node for seg 2

b and c are very close

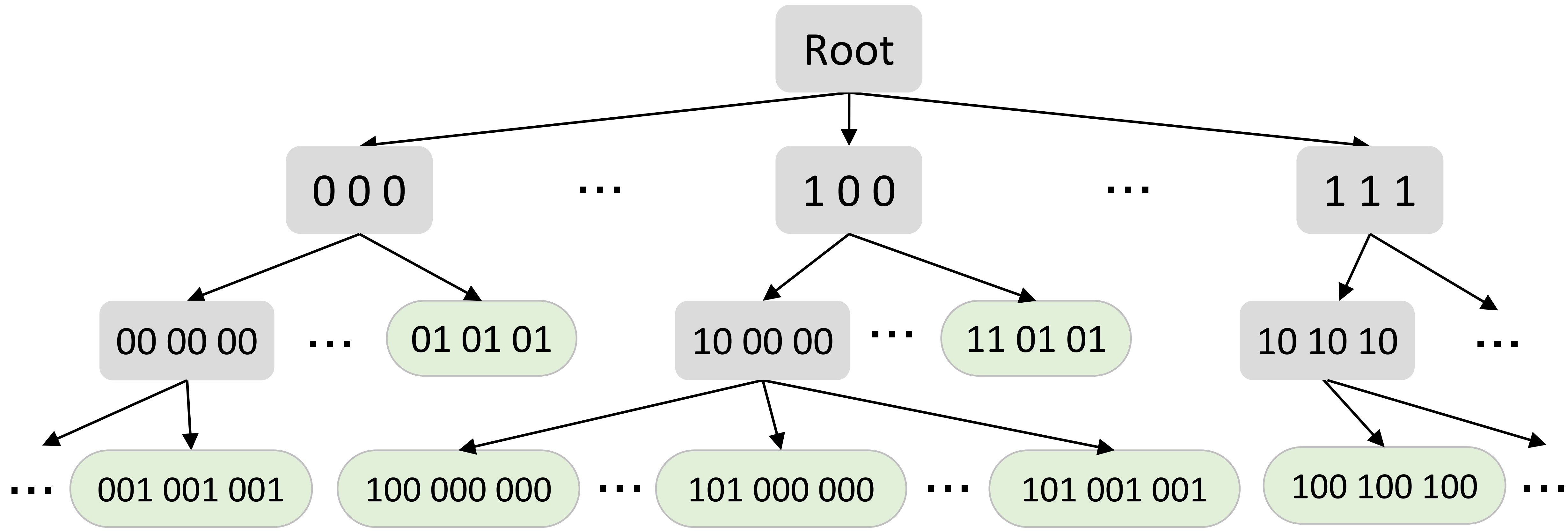
Even Split



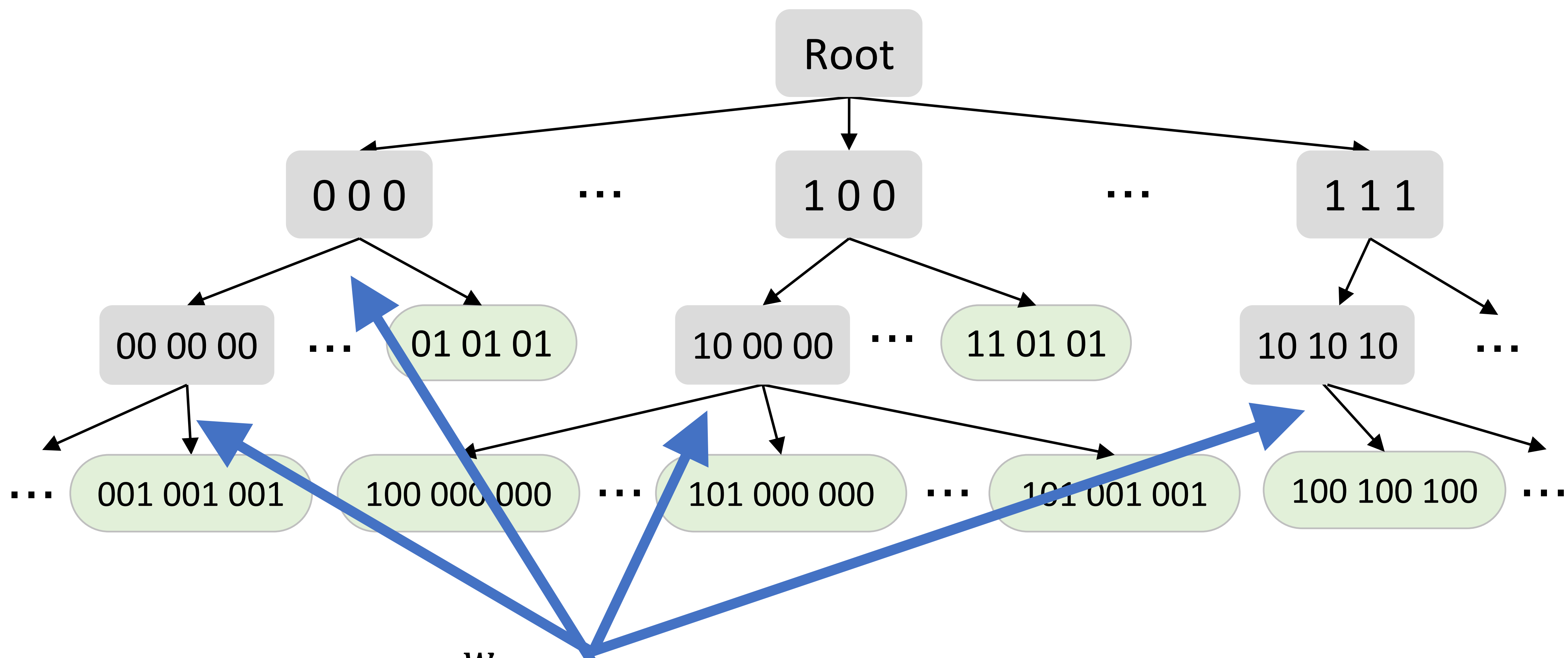
a and b are grouped together in this node

c is excluded

5-Compactness Problem of full-ary iSAX Index



5-Compactness Problem of full-ary iSAX Index



2^w new nodes in each split

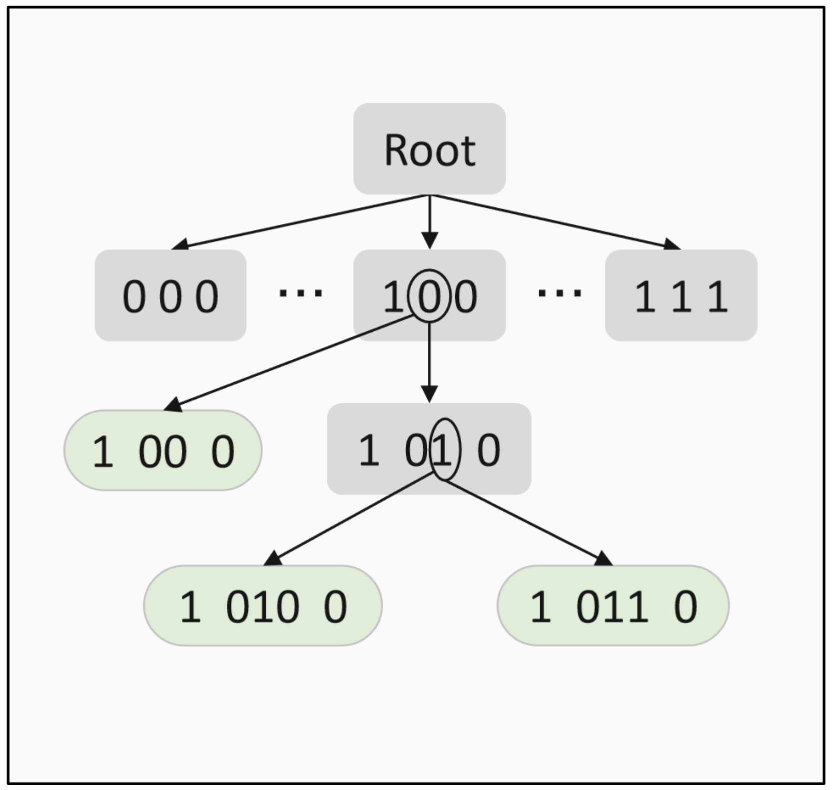
w: number of segments

↓

more random disk accesses in index building and querying!

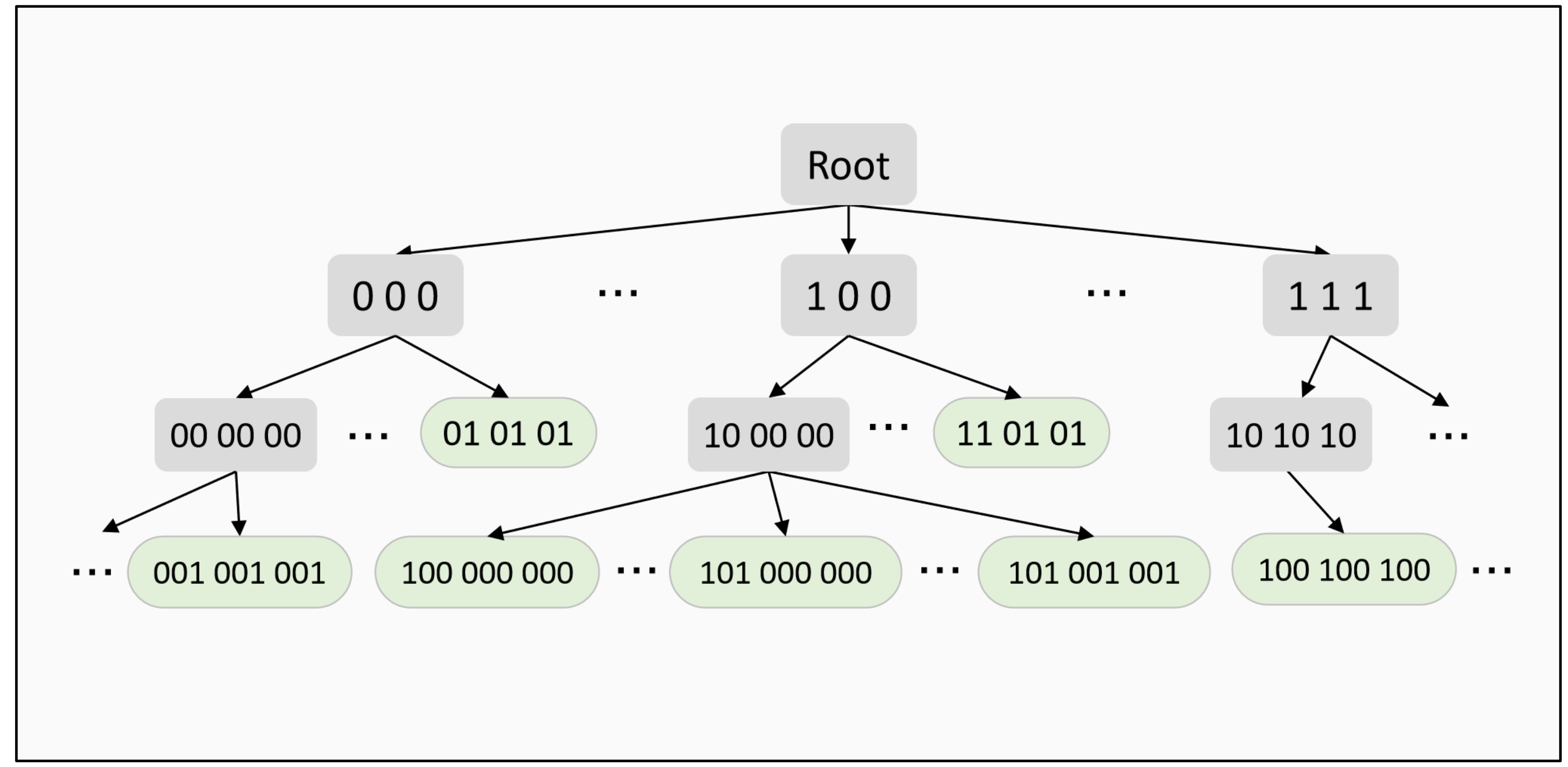
6-Proximity-Compactness Trade-off

2-ary



Splitting Decisions

Full-Ary



Compactness

with proximity problem: skewed split

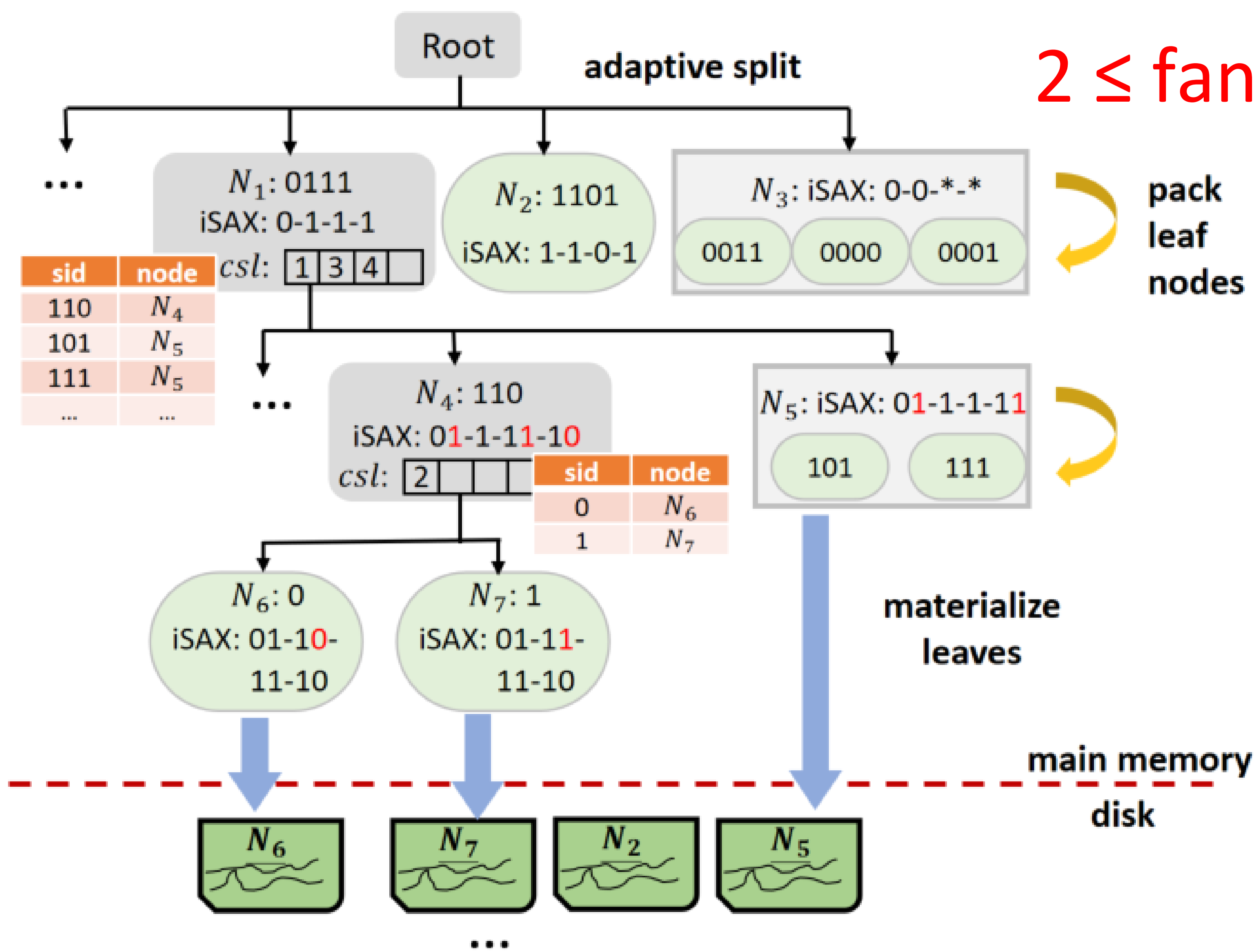
Proximity

with compactness problem: low fill factor

Dumpy dynamically strikes the right balance in this trade-off!

7-Dumpy's Adaptive Split Strategy

Multi-ary Index Structure



$2 \leq \text{fanout} \leq 2^w$

w: number of segments

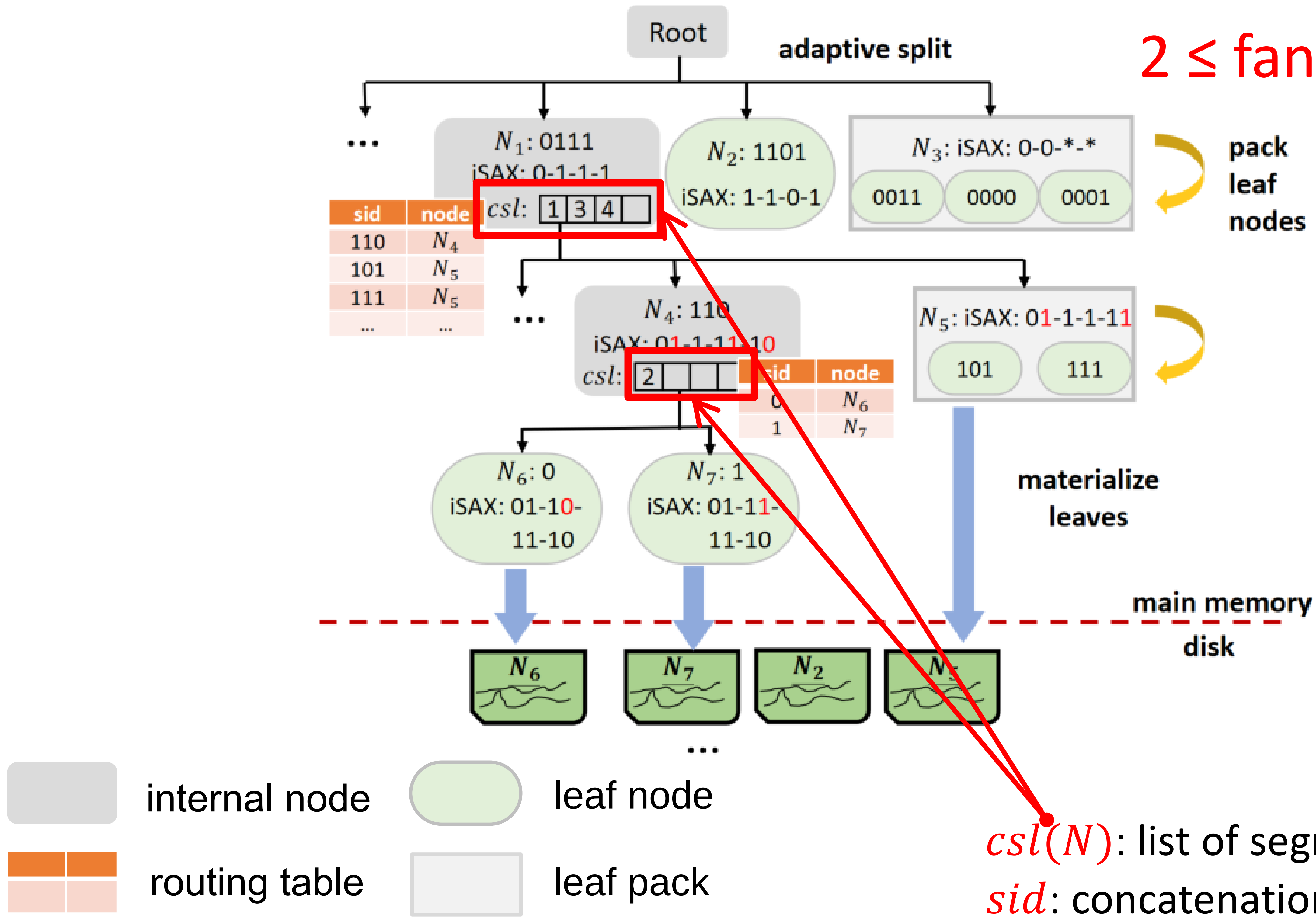
internal node	leaf node
routing table	leaf pack

$csl(N)$: list of segments to be split of node N

sid: concatenation of newly-extended bits compared with parent node

7-Dumpy's Adaptive Split Strategy

Multi-ary Index Structure



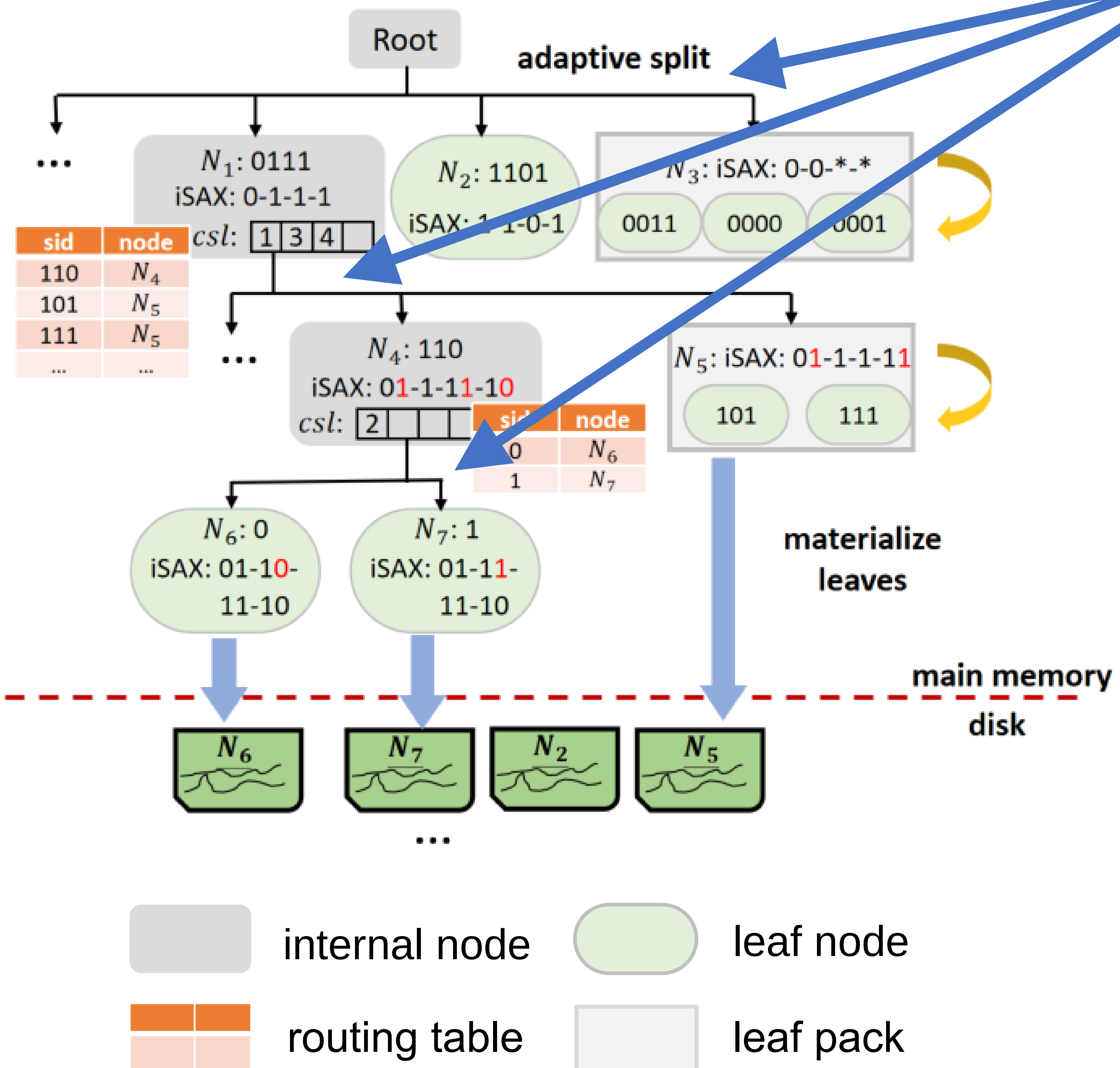
$2 \leq$ fanout $\leq 2^w$

w: number of segments

$csl(N)$: list of segments to be split of node N

sid: concatenation of newly-extended bits compared with parent node

7-Dumpy's Adaptive Split Strategy



Decide fanout and segments to split on-the-fly

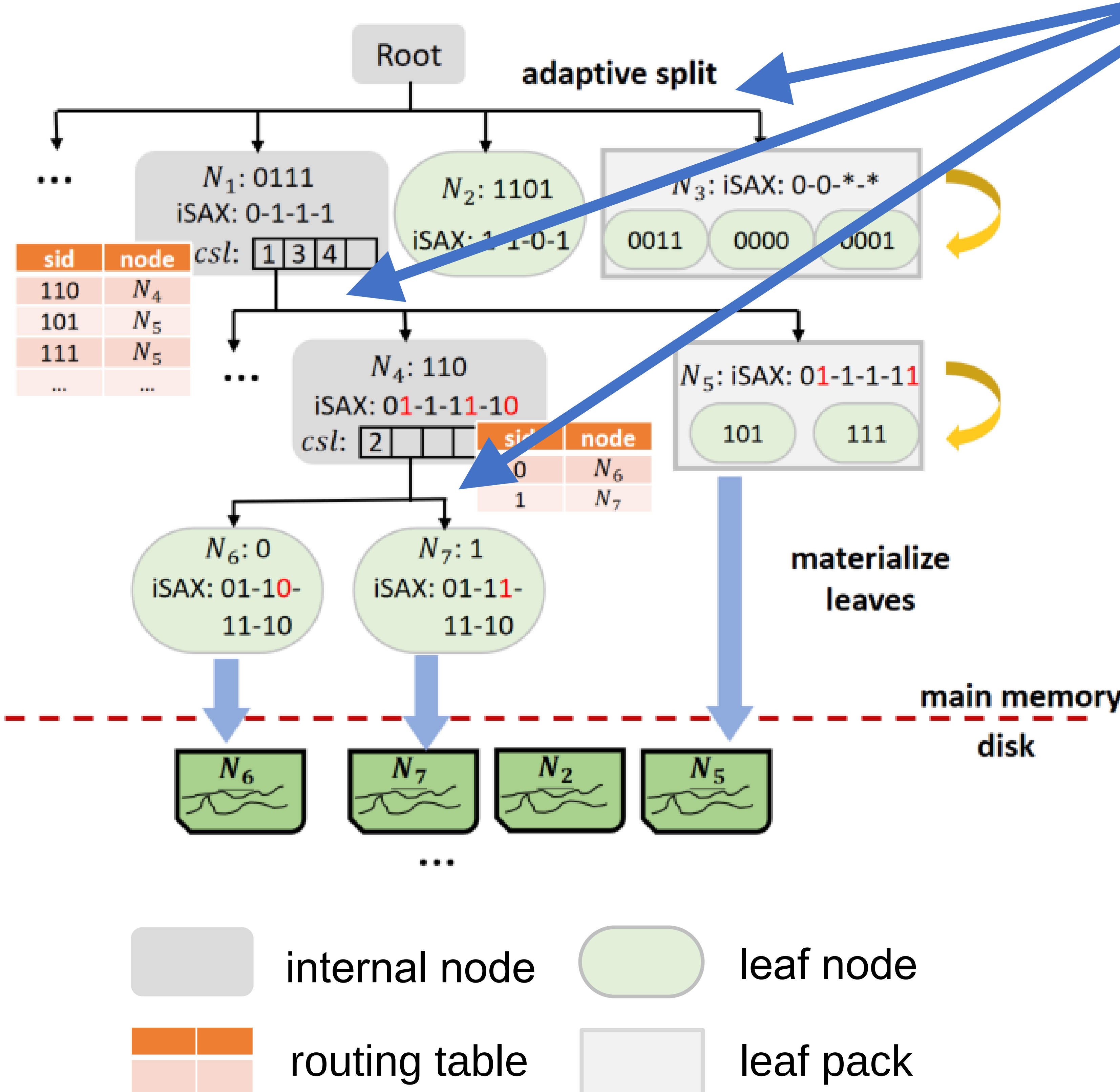
↓

$csl(N)$

$csl(N)$ *can be any non-empty combination of all w segments.*

Totally, $2^w - 1$ *possible ways to split.*
Which one is the optimal?

7-Dumpy's Adaptive Split Strategy



Decide fanout and segments to split on-the-fly

$\boldsymbol{csl(N)}$

$csl(N)$ can be any non-empty combination of all w segments.

Totally, $2^w - 1$ possible ways to split.
Which one is the optimal?

Proximity-Compactness Trade-off

objective function:

$$\max_{csl(N)} e^{\sqrt{\frac{1}{|csl(N)|} Var(X'_N)}} + \alpha * e^{-(1+o)\sigma_F}$$

Proximity term

weight factor

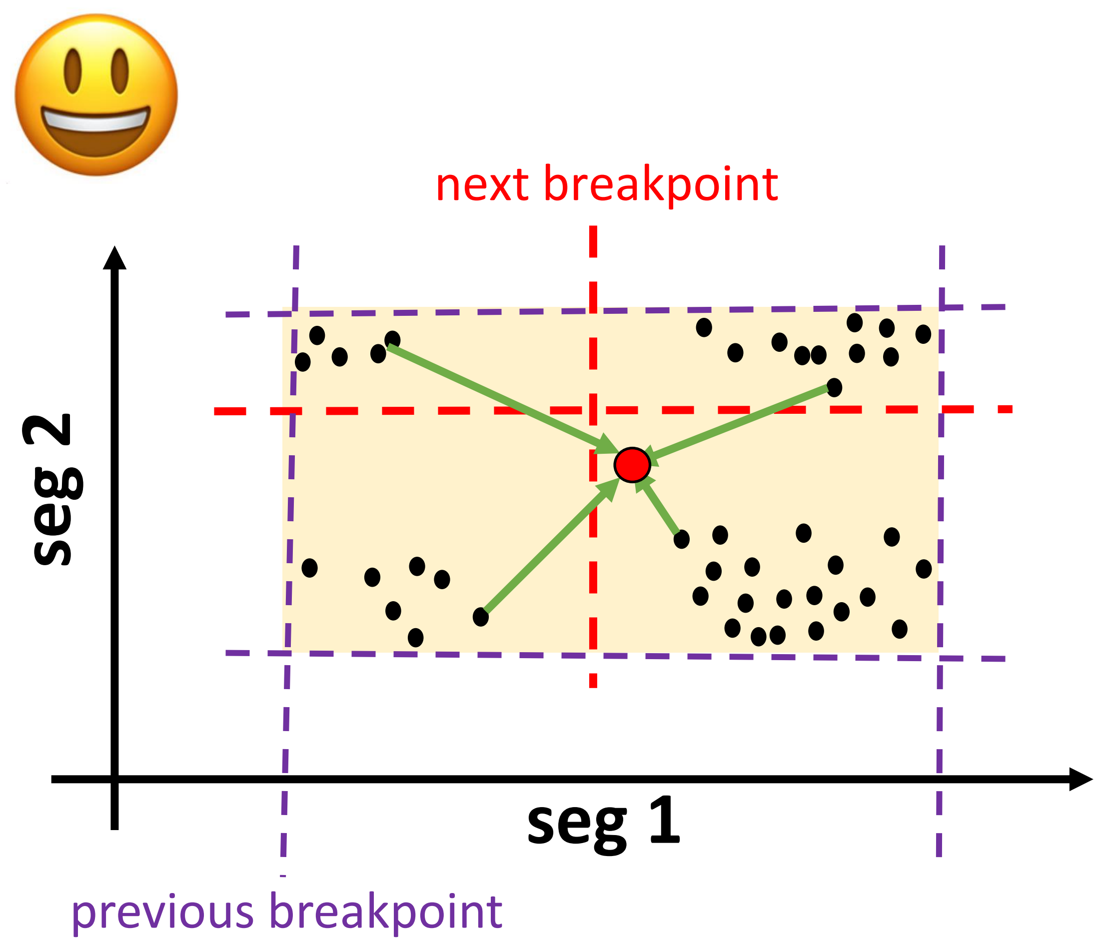
Compactness term

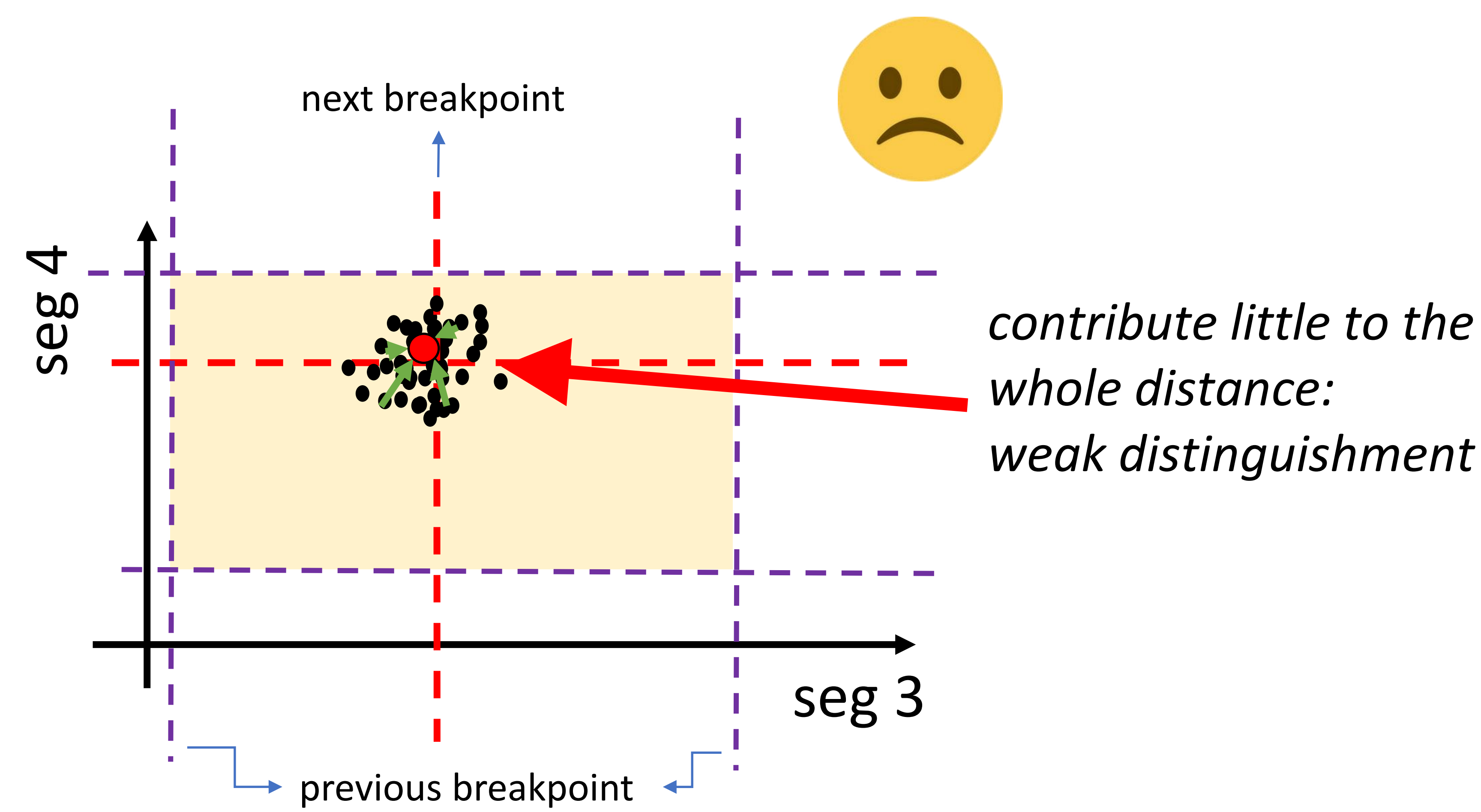
7-Dumpy's Adaptive Split Strategy

Innovation 1: Adaptive split strategy

$$\max_{csl(N)} \boxed{e^{\sqrt{\frac{1}{|csl(N)|}Var(\chi'_N)}}} + \alpha * e^{-(1+o)\sigma_F}$$

Proximity term *avg. distance to the centroid*



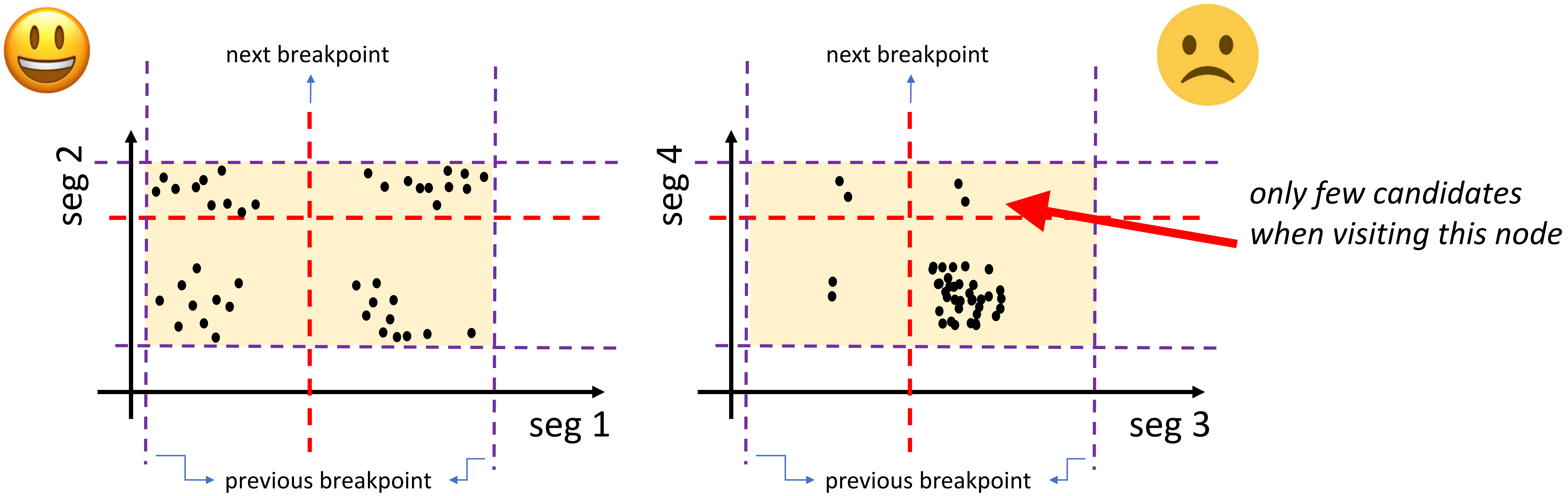


- PAA points projected on the segments of the plan
- $\boldsymbol{\mu}$: centroid of these SAX points

Innovation 1: Adaptive split strategy

$$\max_{csl(N)} e^{\sqrt{\frac{1}{|csl(N)|} Var(\chi'_N)}} + \alpha * \boxed{e^{-(1+o)\sigma_F}}$$

Compactness term *stdev of fill factors of child nodes*



- PAA points projected on the segments of the plan

8-Fast search for optimal split

1. pre-compute variance

$$Var(\mathcal{X}'_N) = \sum_{cs \in csl(N)} Var(\Pi_{cs}(\mathcal{X}_N))$$

ONE time scan for all split plans

8-Fast search for optimal split

1. pre-compute variance

$$Var(\mathcal{X}'_N) = \sum_{cs \in csl(N)} Var(\Pi_{cs}(\mathcal{X}_N))$$

ONE time scan for all split plans

2. restrict the search space

$$\max(1, \log \frac{c_N}{F_r * th}) \leq |csl(N)| \leq \min(w, \log \frac{c_N}{F_l * th})$$

skip the split plan whose fanout is too large or small

8-Fast search for optimal split

1. pre-compute variance

$$Var(X'_N) = \sum_{cs \in csl(N)} Var(\Pi_{cs}(X_N))$$

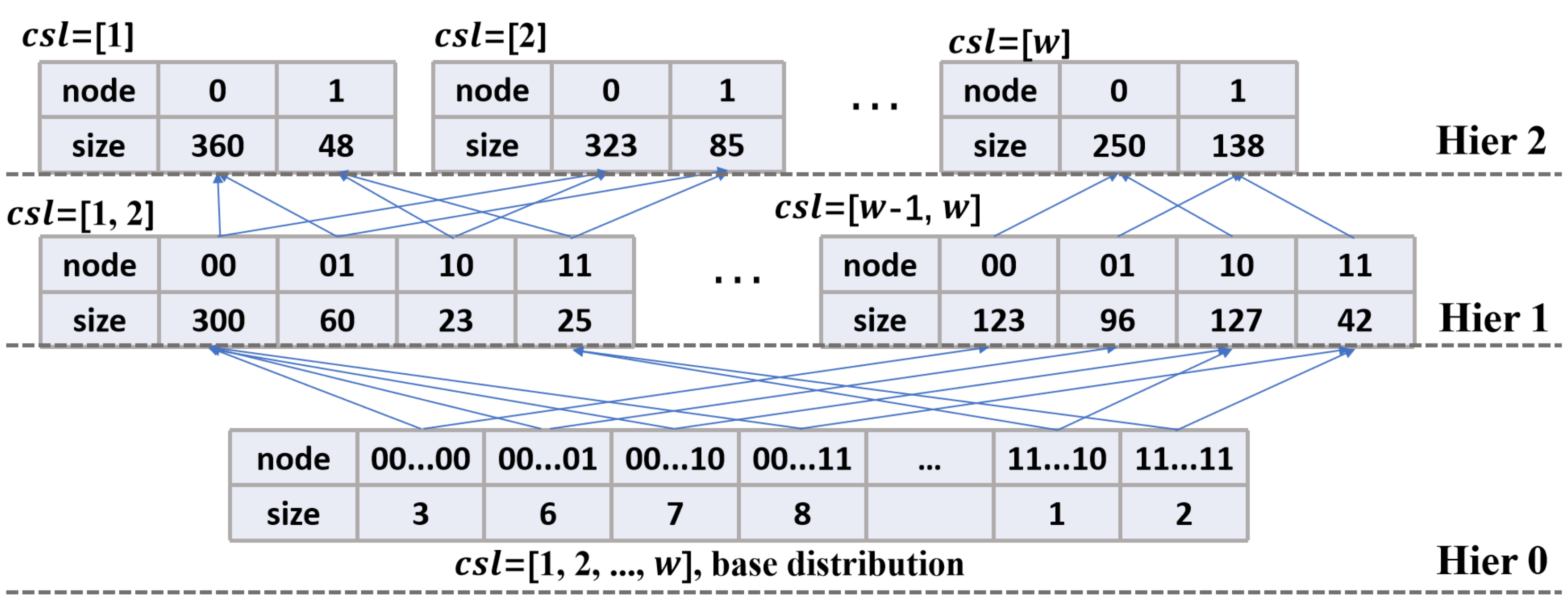
ONE time scan for all split plans

2. restrict the search space

$$\max(1, \log \frac{c_N}{F_r * th}) \le |csl(N)| \le \min(w, \log \frac{c_N}{F_l * th})$$

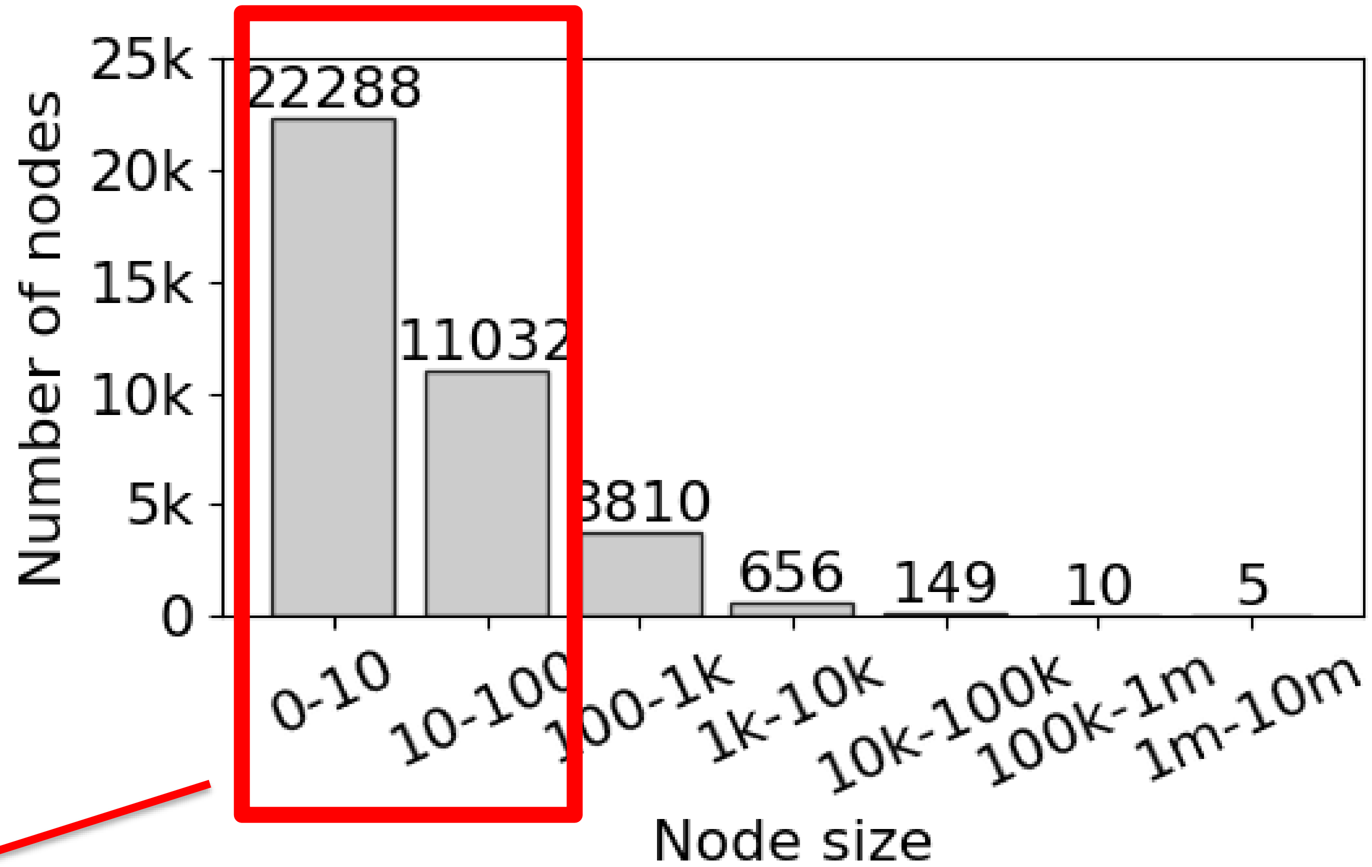
skip the split plan whose fanout is too large or small

3. hierarchically compute sizes of child nodes



recursive computing:
CONSTANT complexity for most split plans

9-Data Skewness of iSAX Index

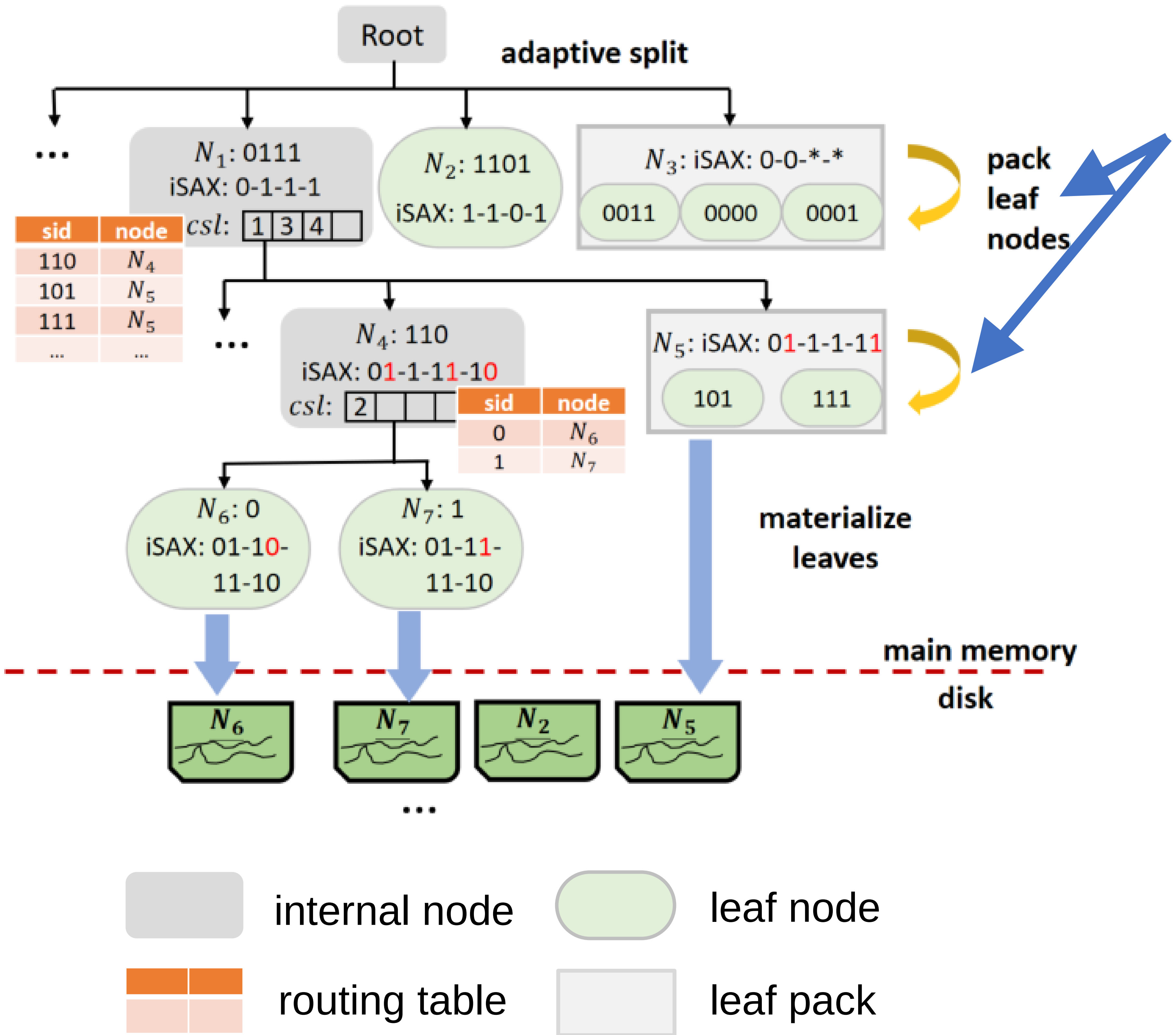


Most leaves are very small!

Node size distribution on the 1st layer of iSAX (w=16)

w: number of segments

10-Dumpy's Leaf Node Packing

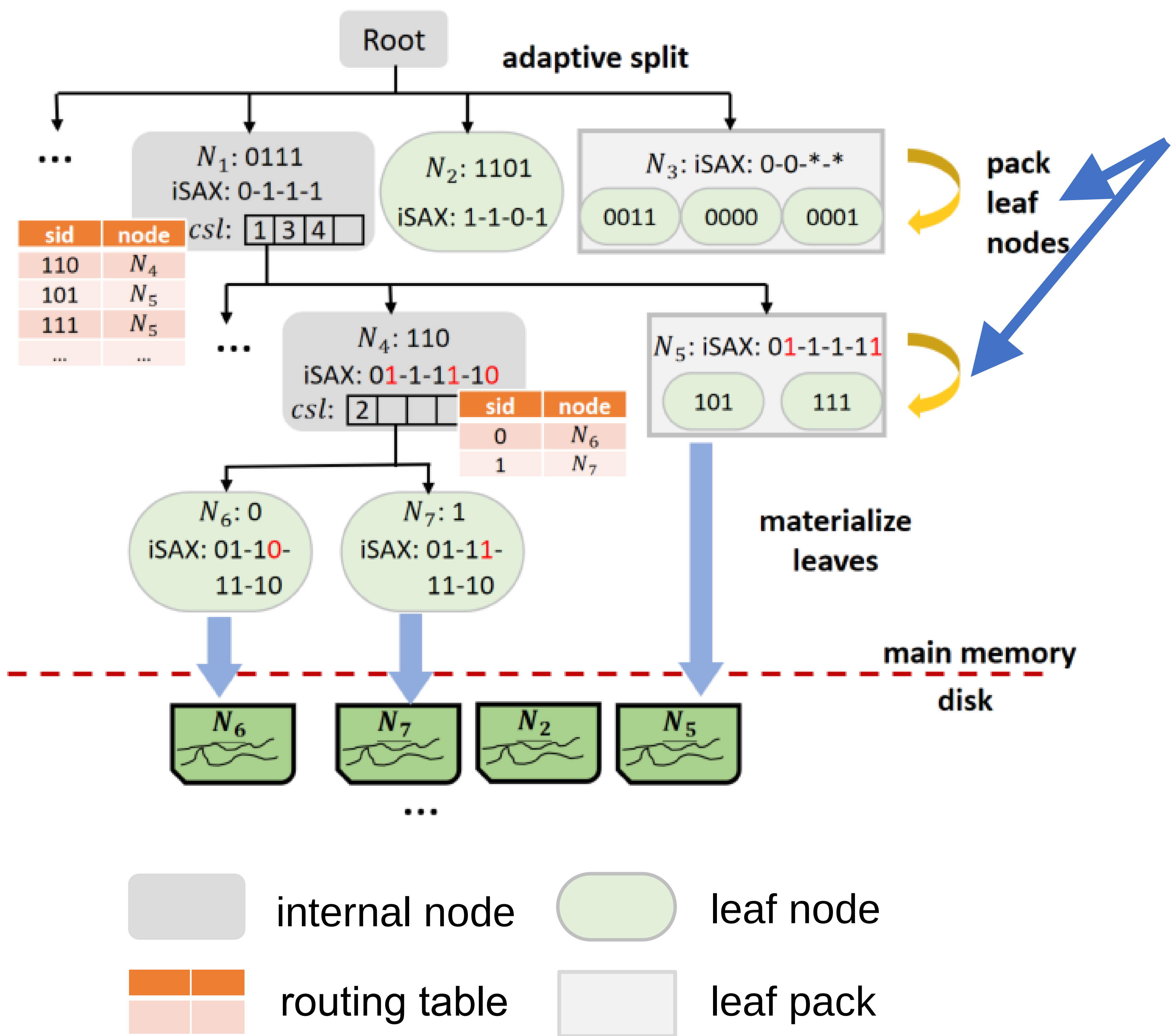


Group small similar leaf nodes into a pack

Input: small leaf nodes under the same parent (siblings)

Output: A group of leaf packs within the size constraints

10-Dumpy's Leaf Node Packing



Group small close leaf nodes into a pack

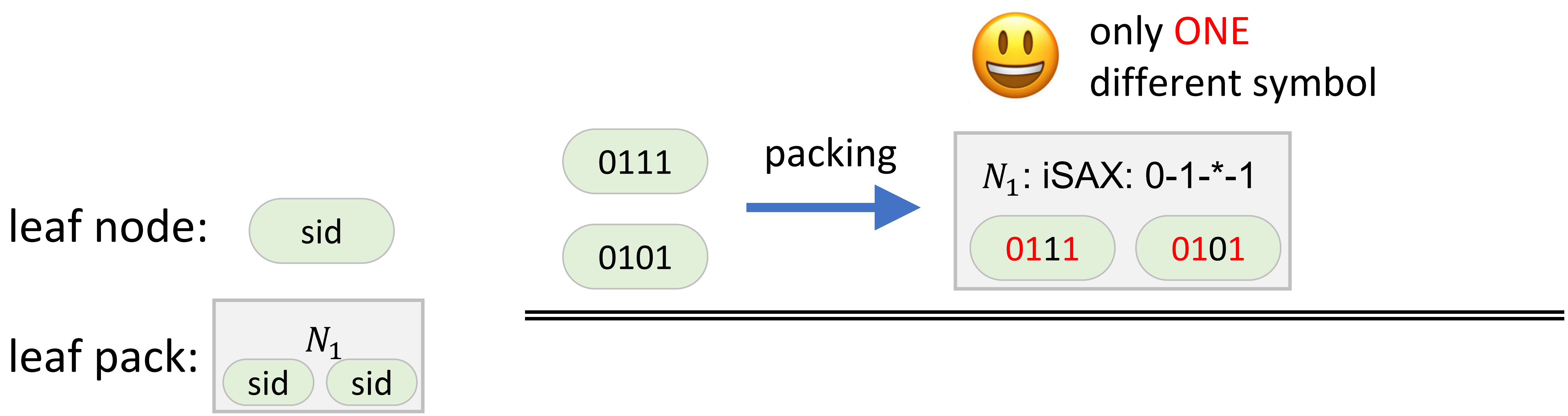
How to measure?

Input: small leaf nodes under the same parent (siblings)

Output: A group of leaf packs within the size constraints

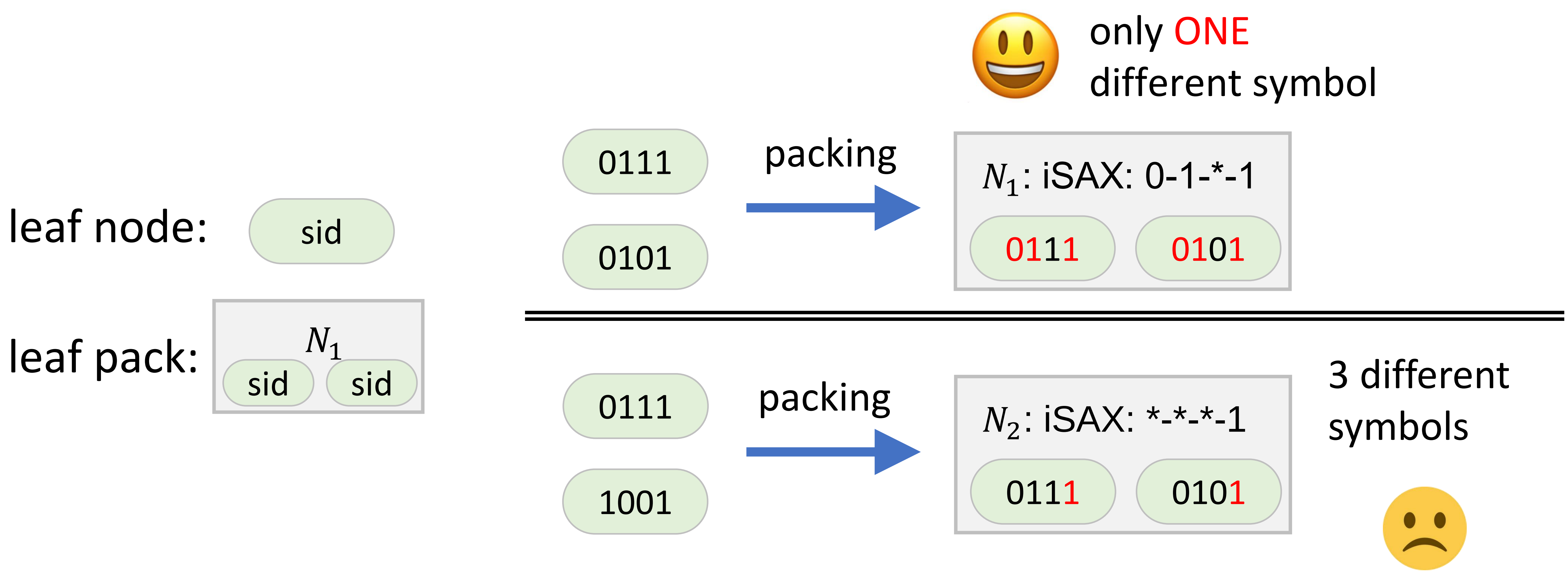
10-Dumpy's Leaf Node Packing

- Input: <u>Small</u> leaf nodes under the same parent (siblings)
- Output: A group of leaf node packs within the size constriant



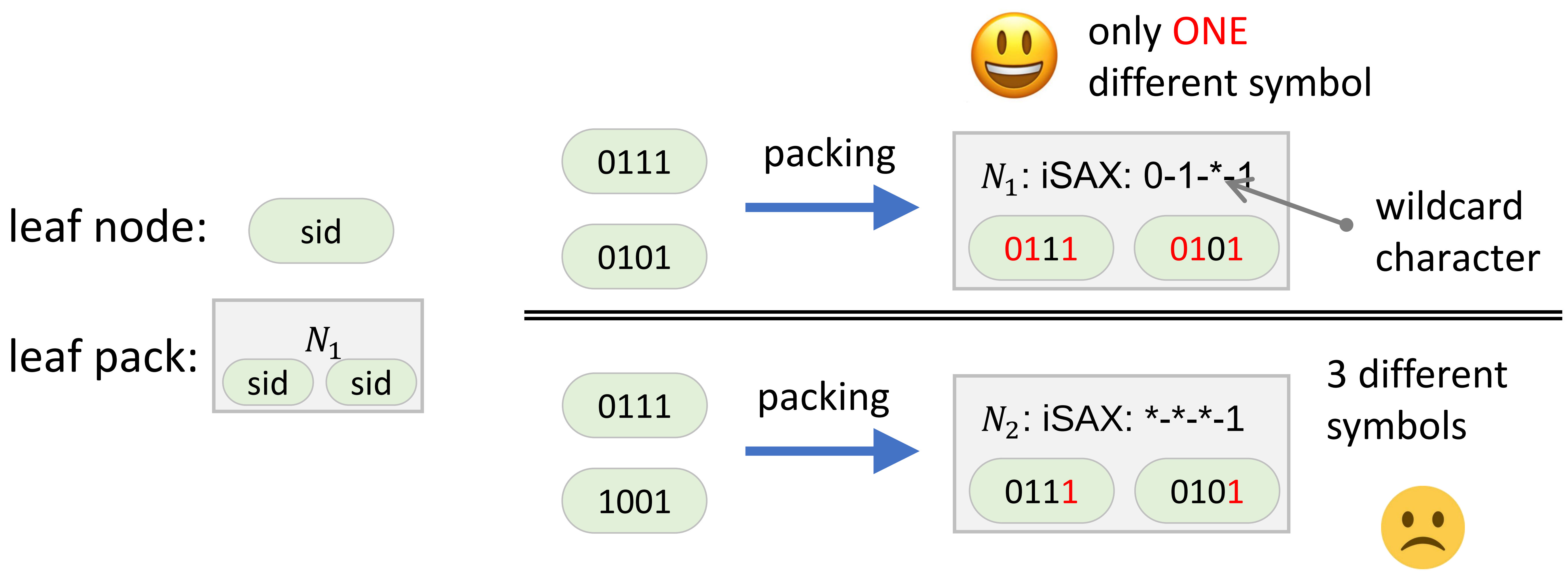
10-Dumpy's Leaf Node Packing

- Input: Small leaf nodes under the same parent (siblings)
- Output: A group of leaf node packs within the size constriant



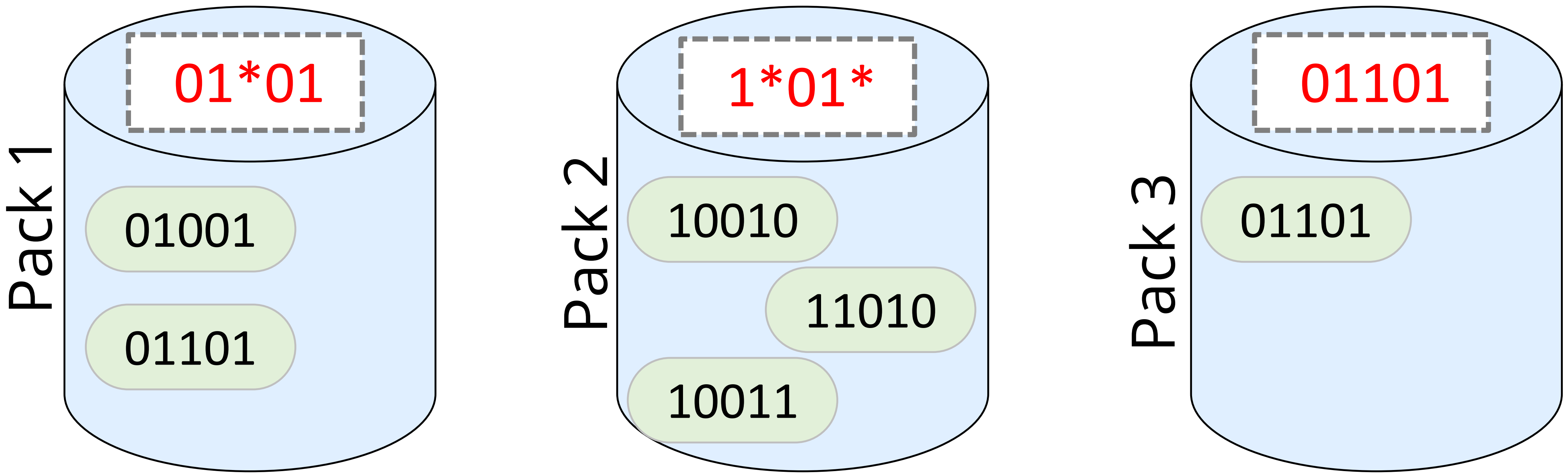
10-Dumpy's Leaf Node Packing

- Input: Small leaf nodes under the same parent (siblings)
- Output: A group of leaf node packs within the size constriant
- Core idea: **limit the number of different symbols (#(*)) in the leaf pack.**



10-Dumpy's Leaf Node Packing

Greedily select the best pack or create a new one

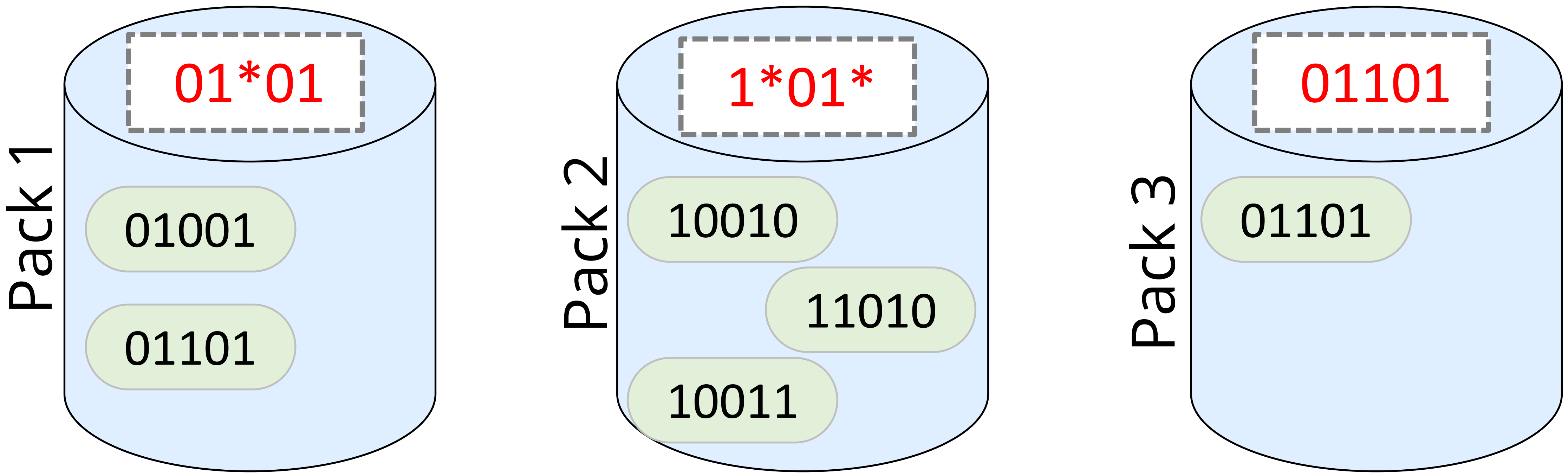


10-Dumpy's Leaf Node Packing

Greedily select the best pack or create a new one

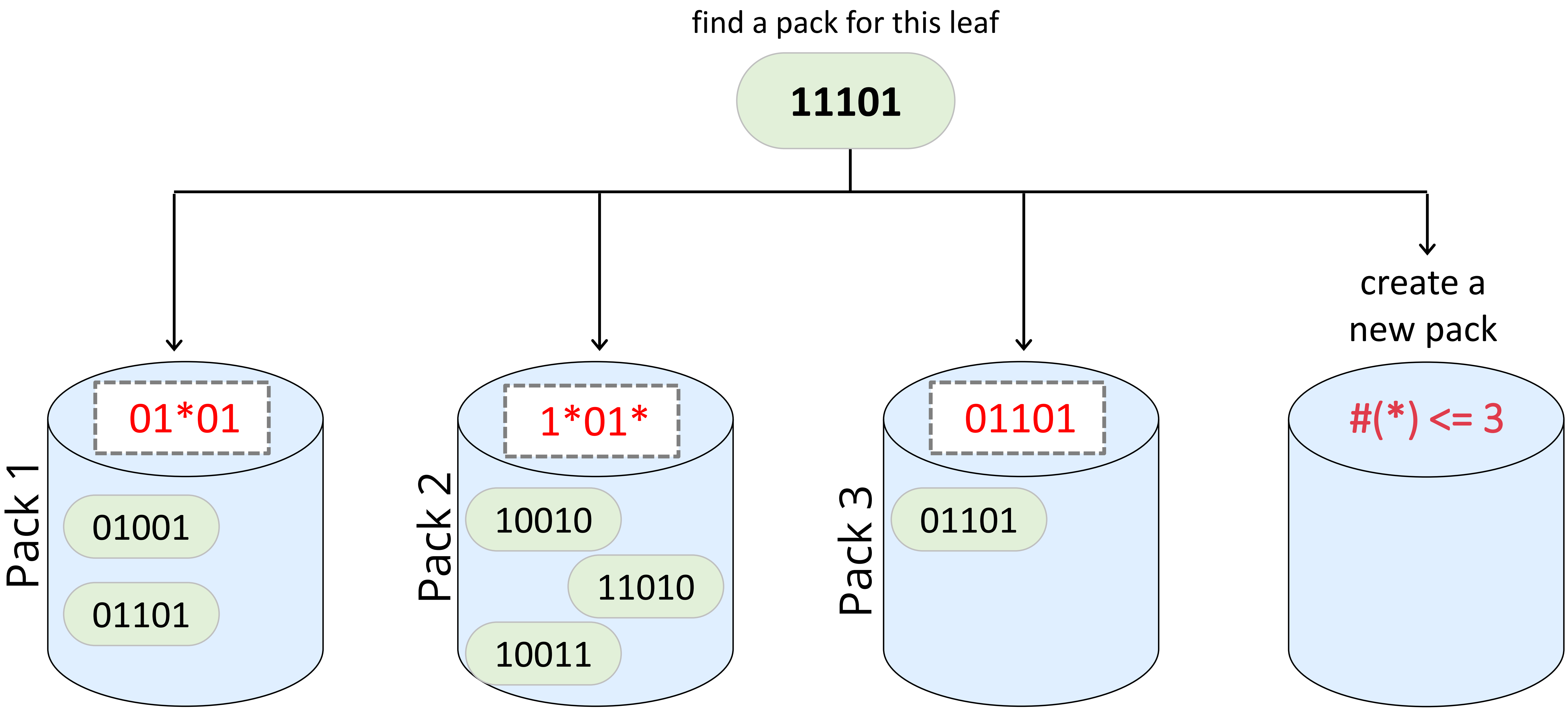
find a pack for this leaf

11101



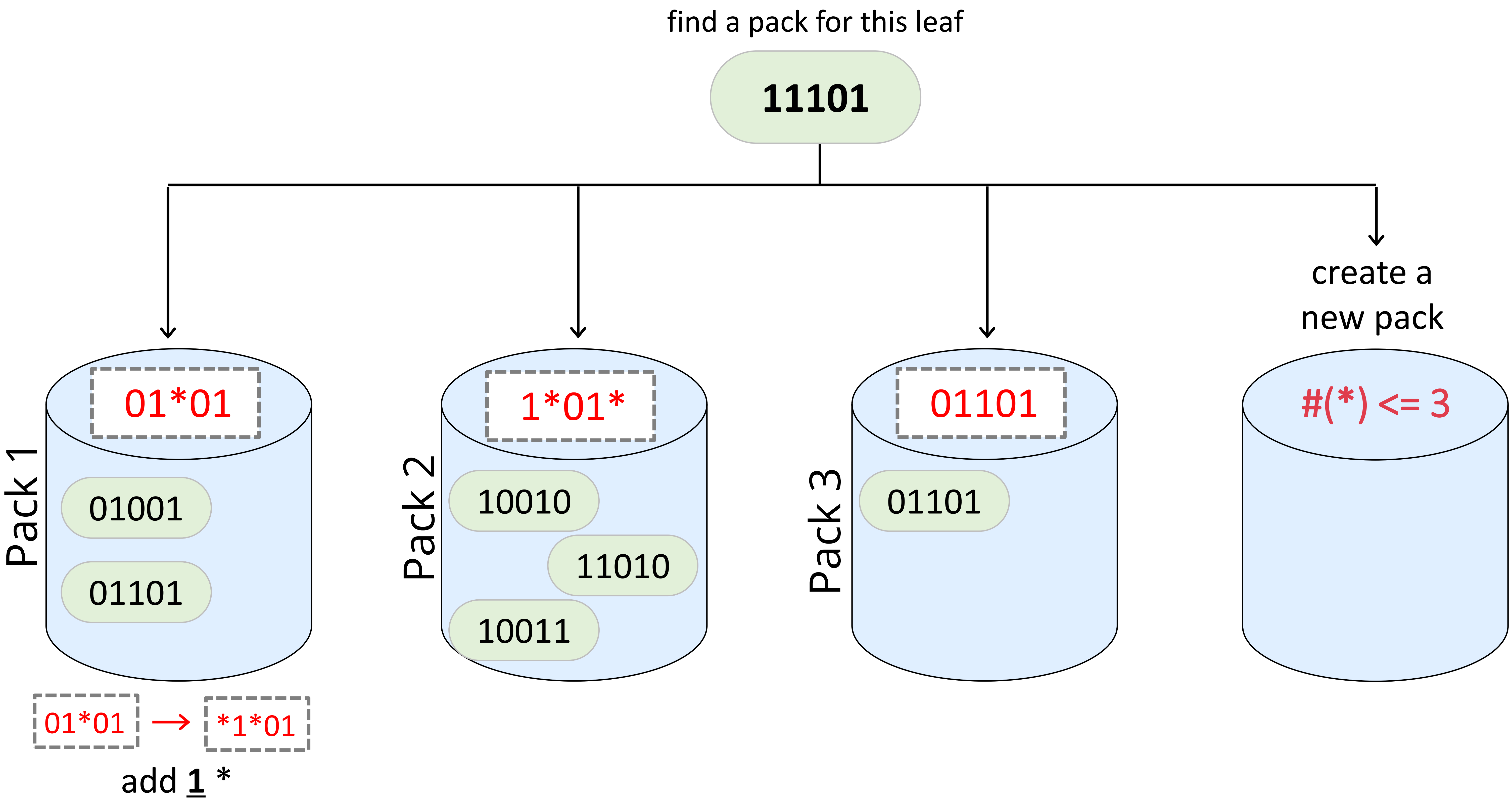
10-Dumpy's Leaf Node Packing

Greedily select the best pack or create a new one



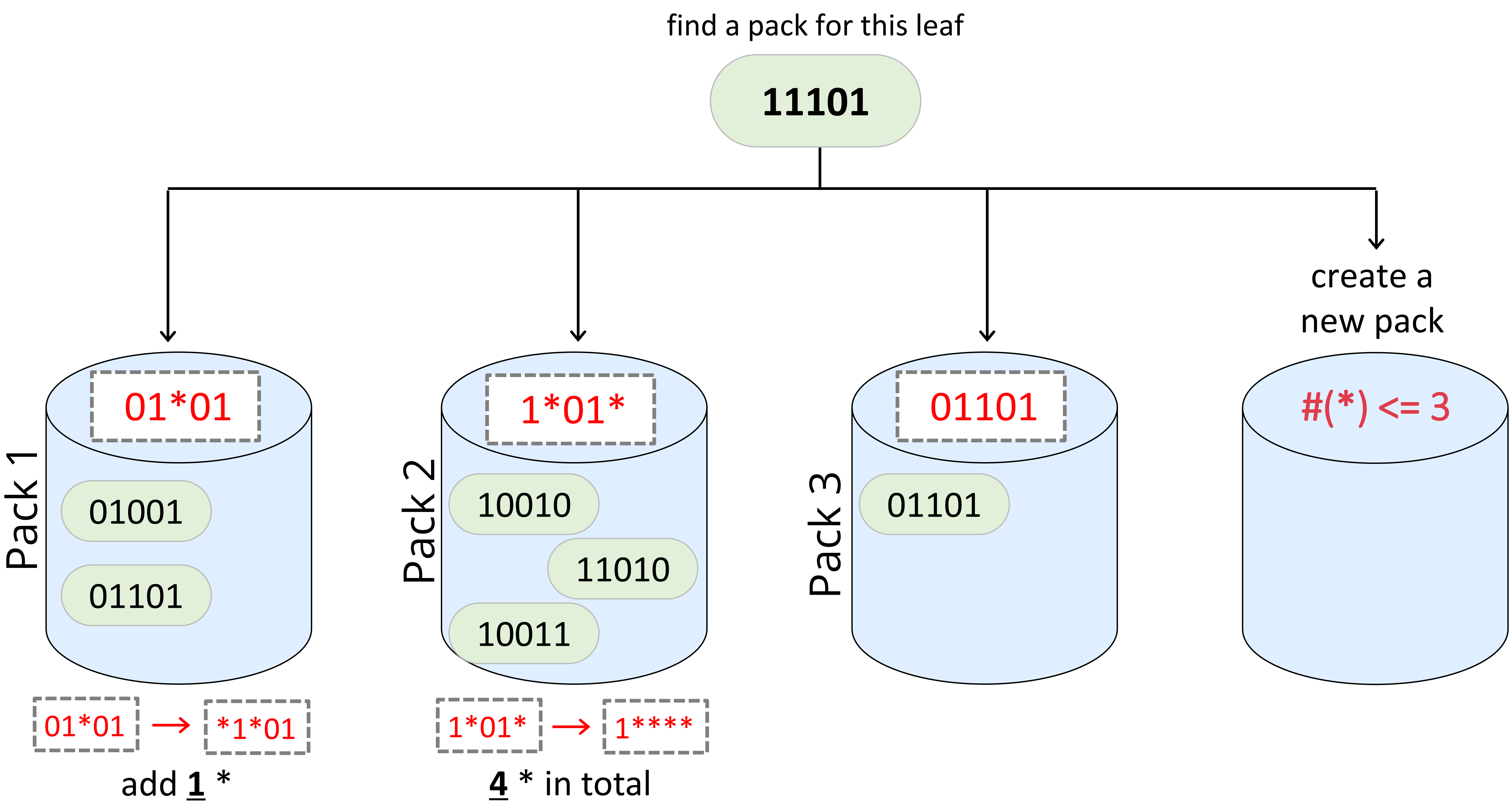
10-Dumpy's Leaf Node Packing

Greedily select the best pack or create a new one



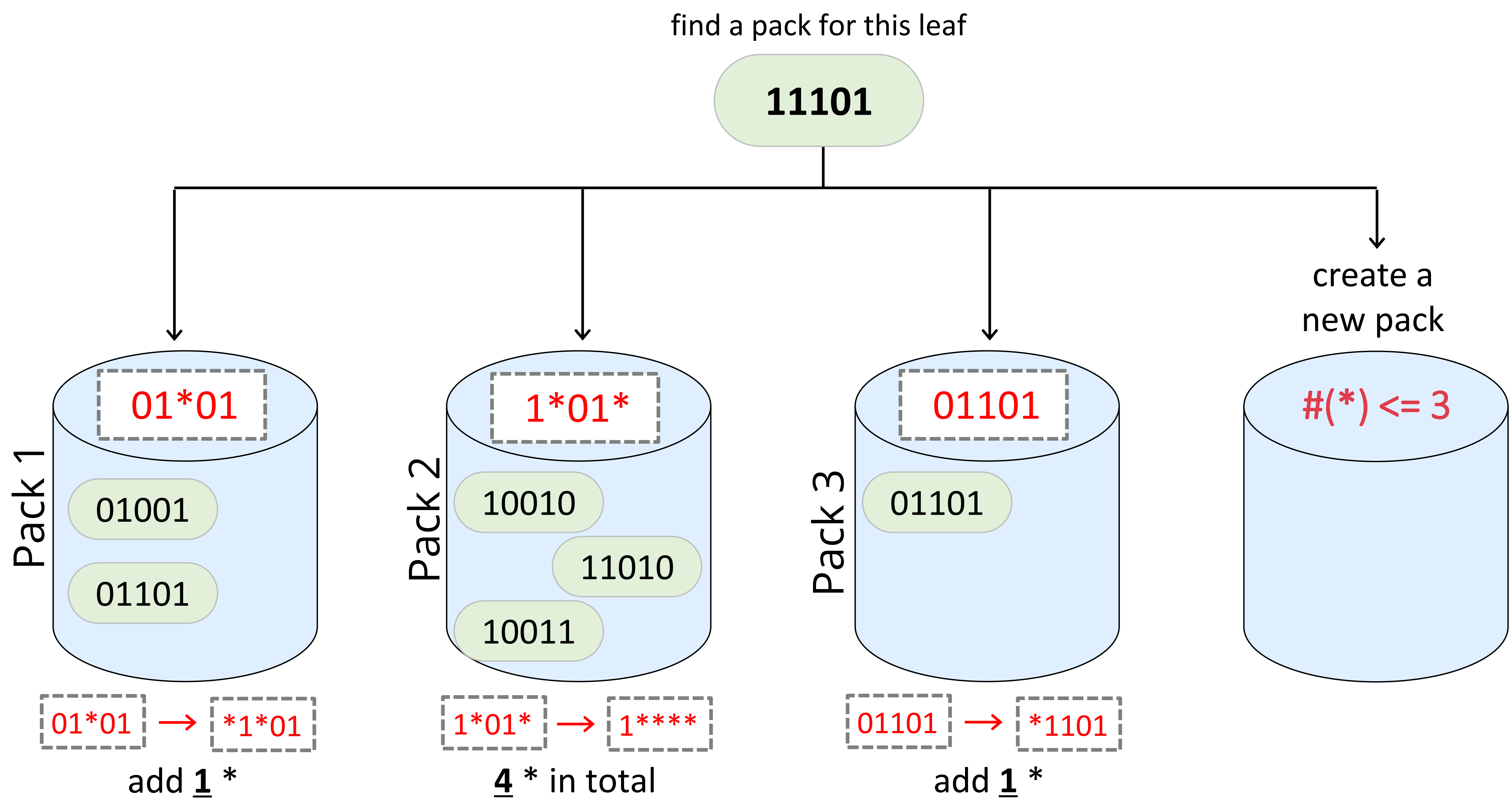
10-Dumpy's Leaf Node Packing

Greedily select the best pack or create a new one



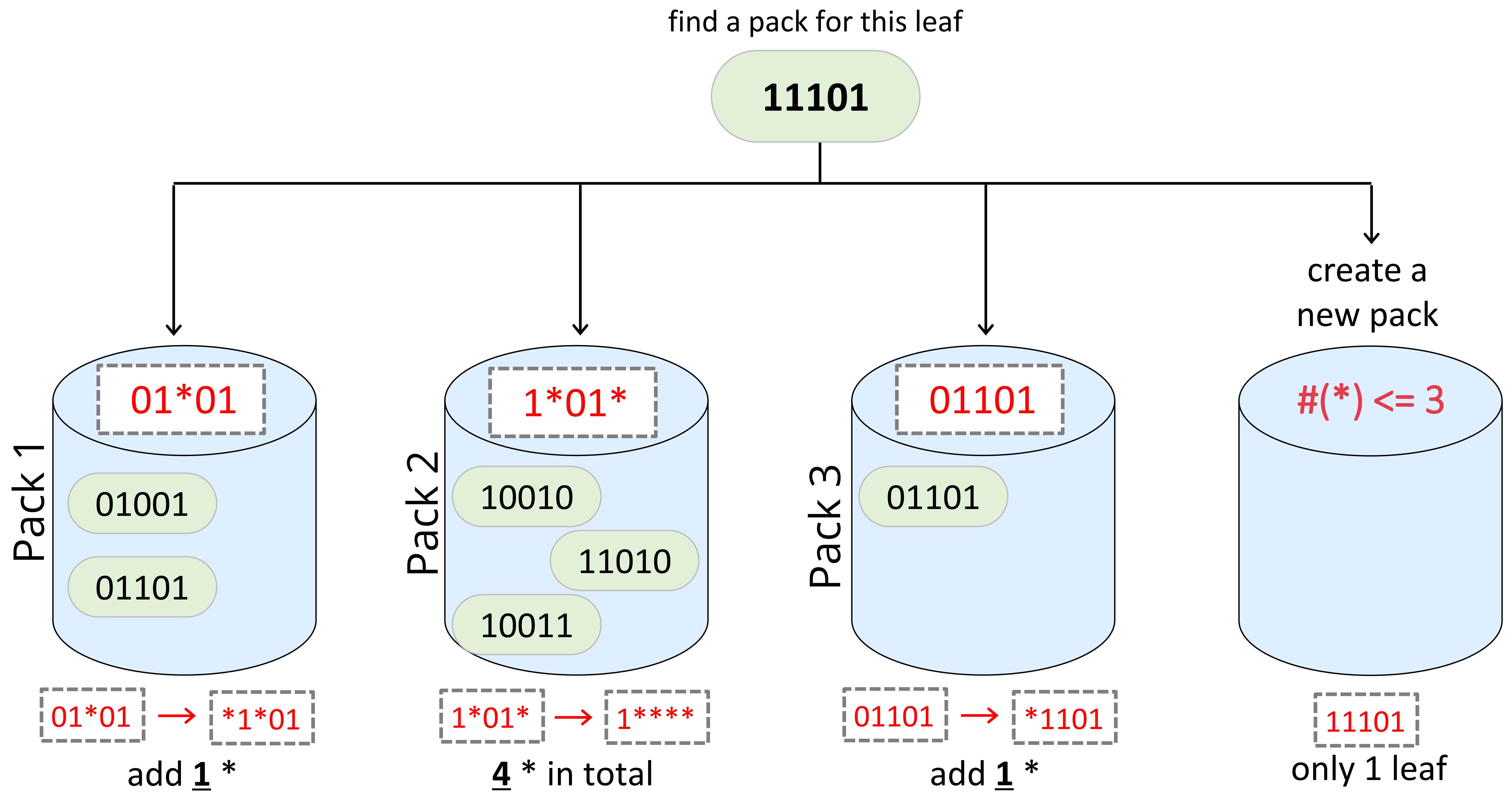
10-Dumpy's Leaf Node Packing

Greedily select the best pack or create a new one



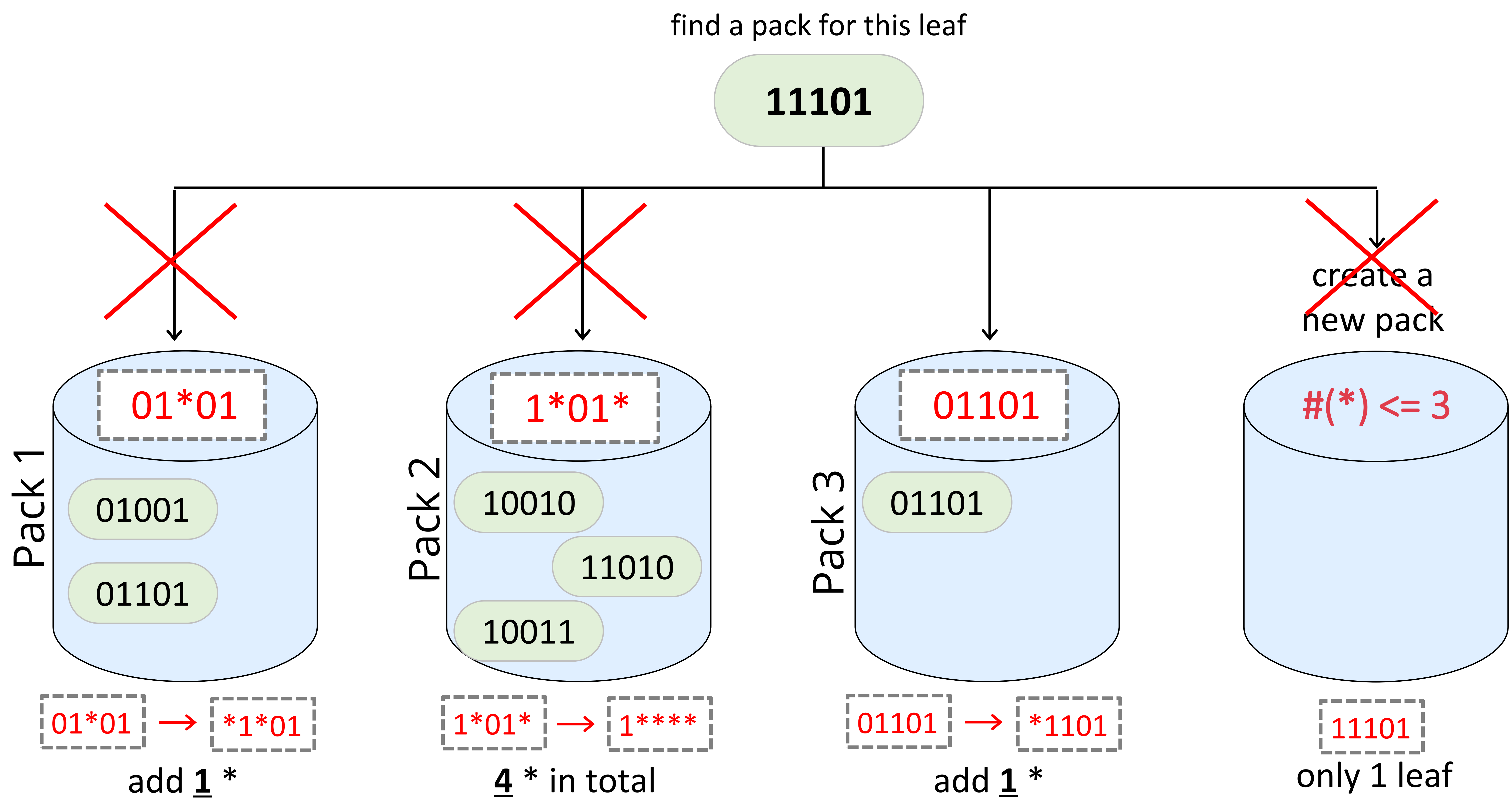
10-Dumpy's Leaf Node Packing

Greedily select the best pack or create a new one



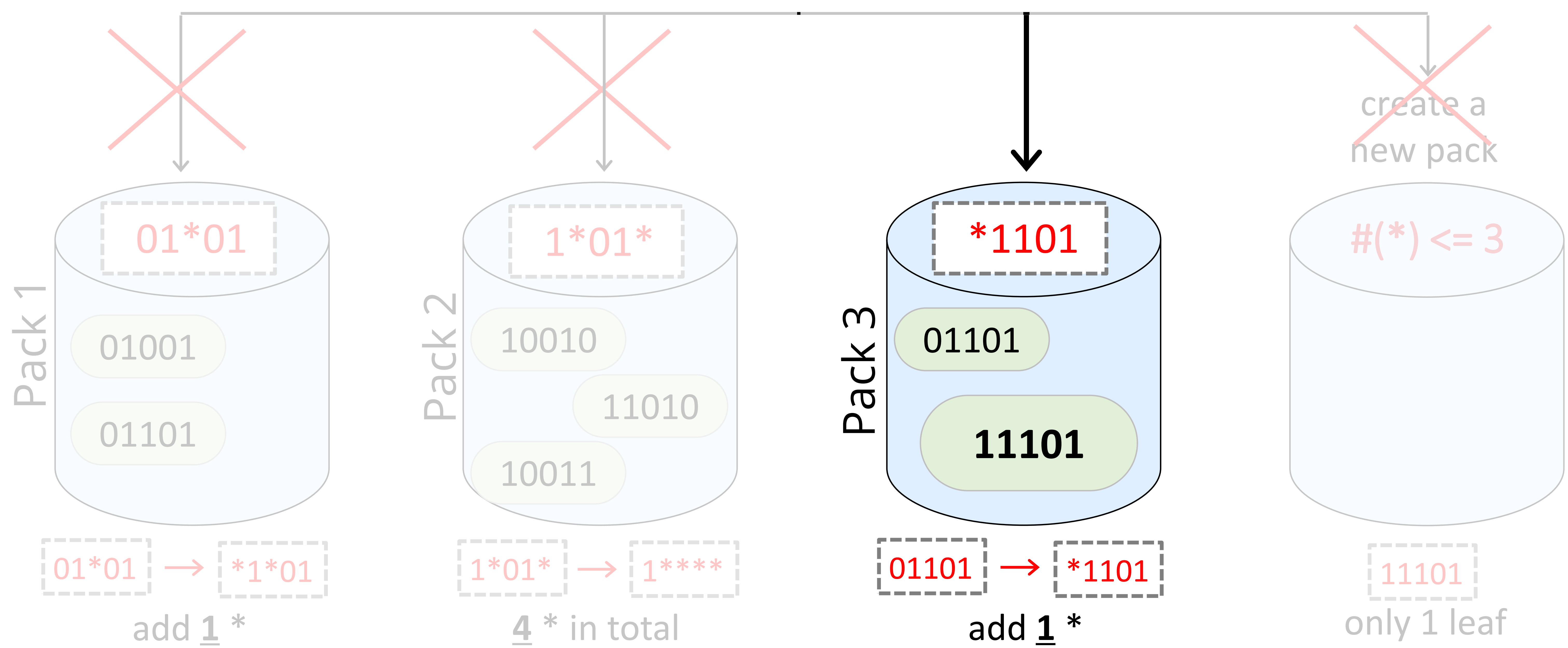
10-Dumpy's Leaf Node Packing

Greedily select the best pack or create a new one



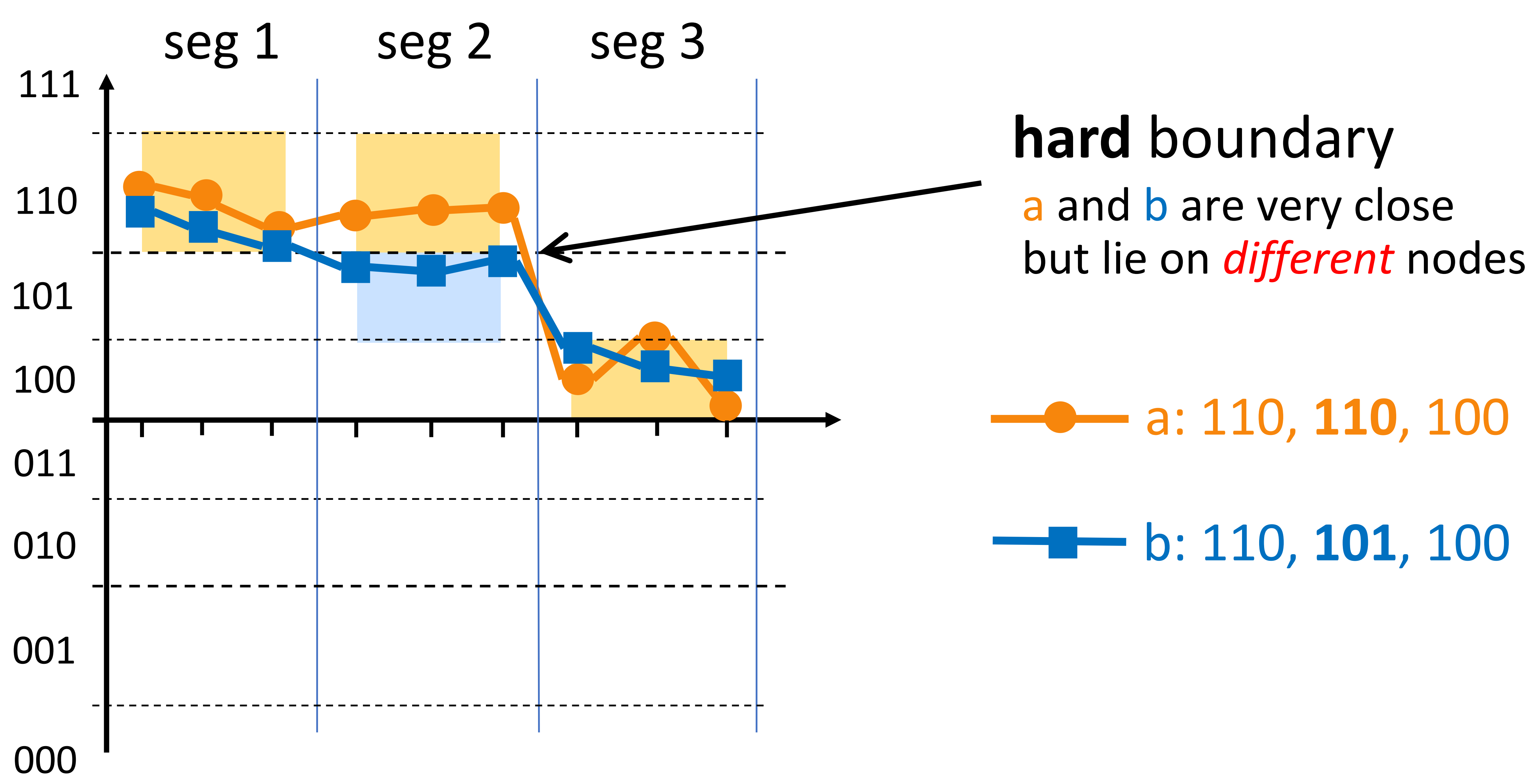
10-Dumpy's Leaf Node Packing

Greedily select the best pack or create a new one



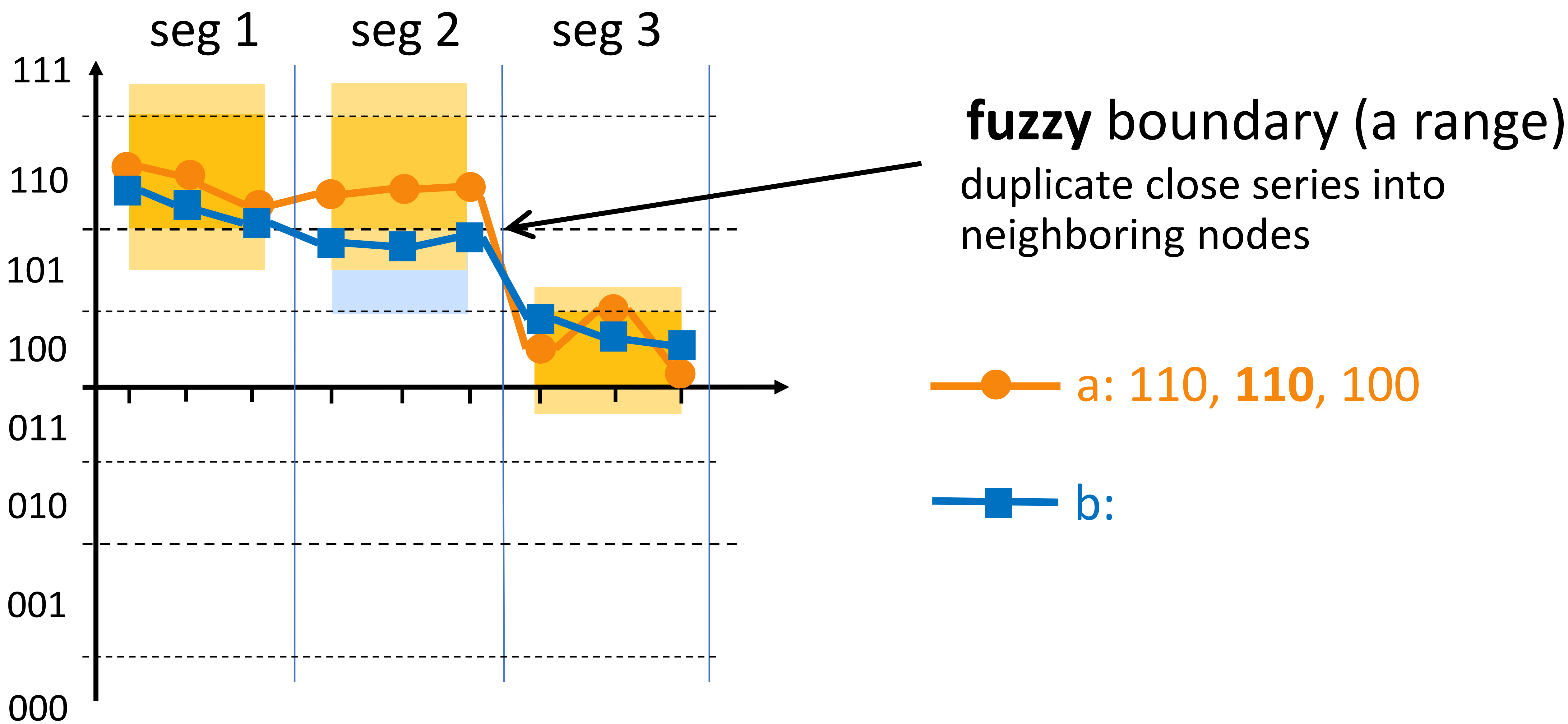
11-Dumpy-Fuzzy

An example of *Boundary Issue*:



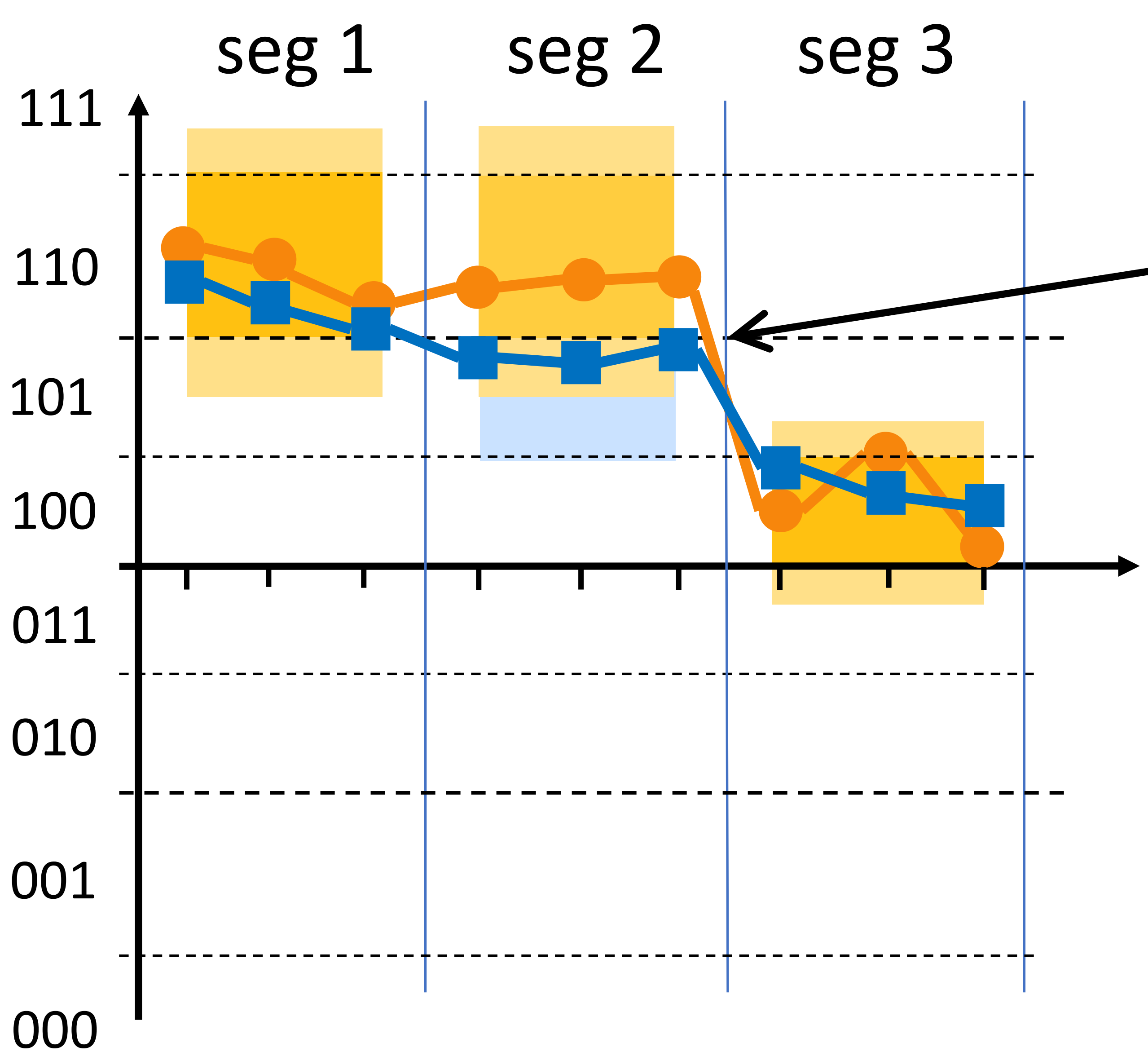
11-Dumpy-Fuzzy

An example of ***Boundary Issue***:



11-Dumpy-Fuzzy

An example of ***Boundary Issue***:



fuzzy boundary (a range)
duplicate close series into neighboring nodes

a: 110, **110**, 100

b: 110, **110**, 100
110, **101**, 100

belongs to **2** leaves

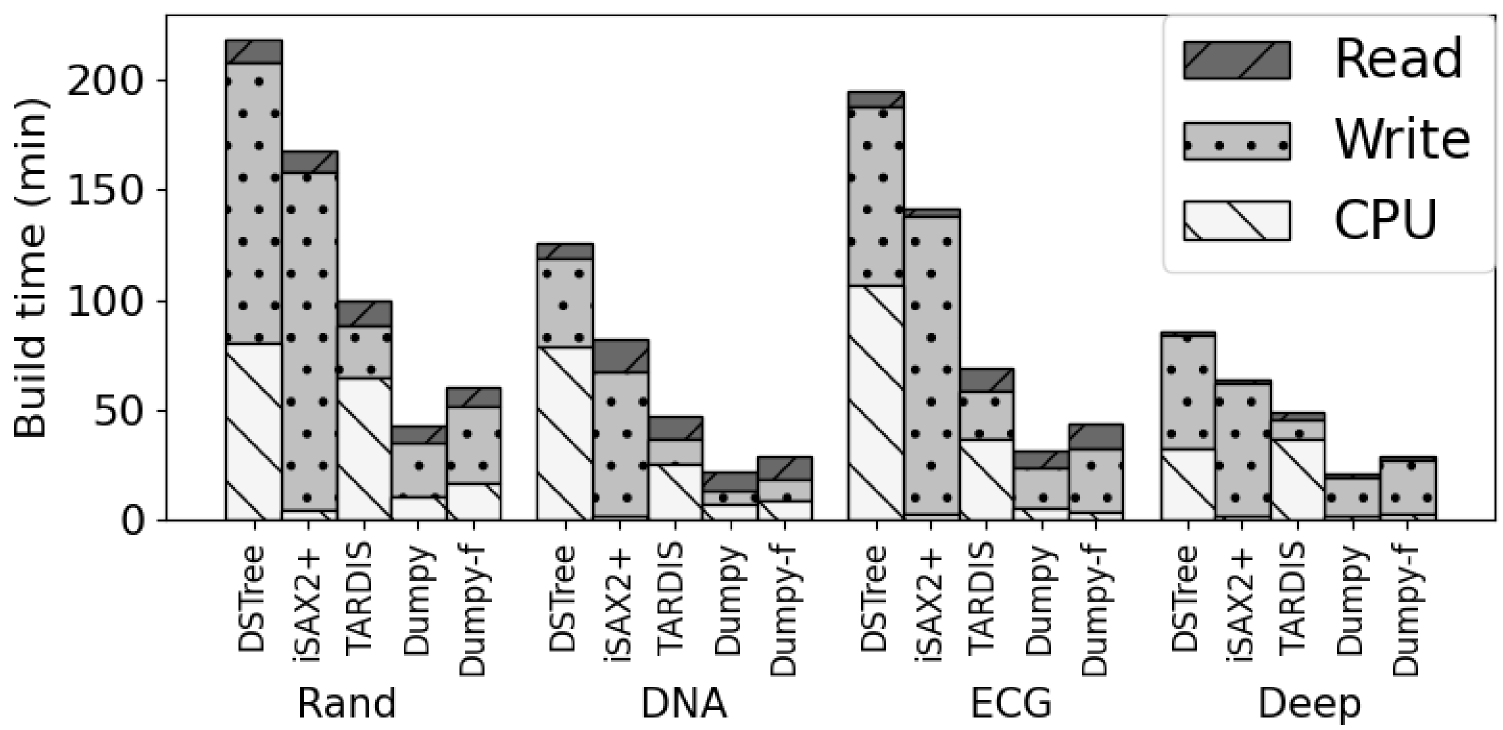
12-Experimental Setup

- Datasets
  - synthetic: Rand
  - real: DNA, ECG, Deep

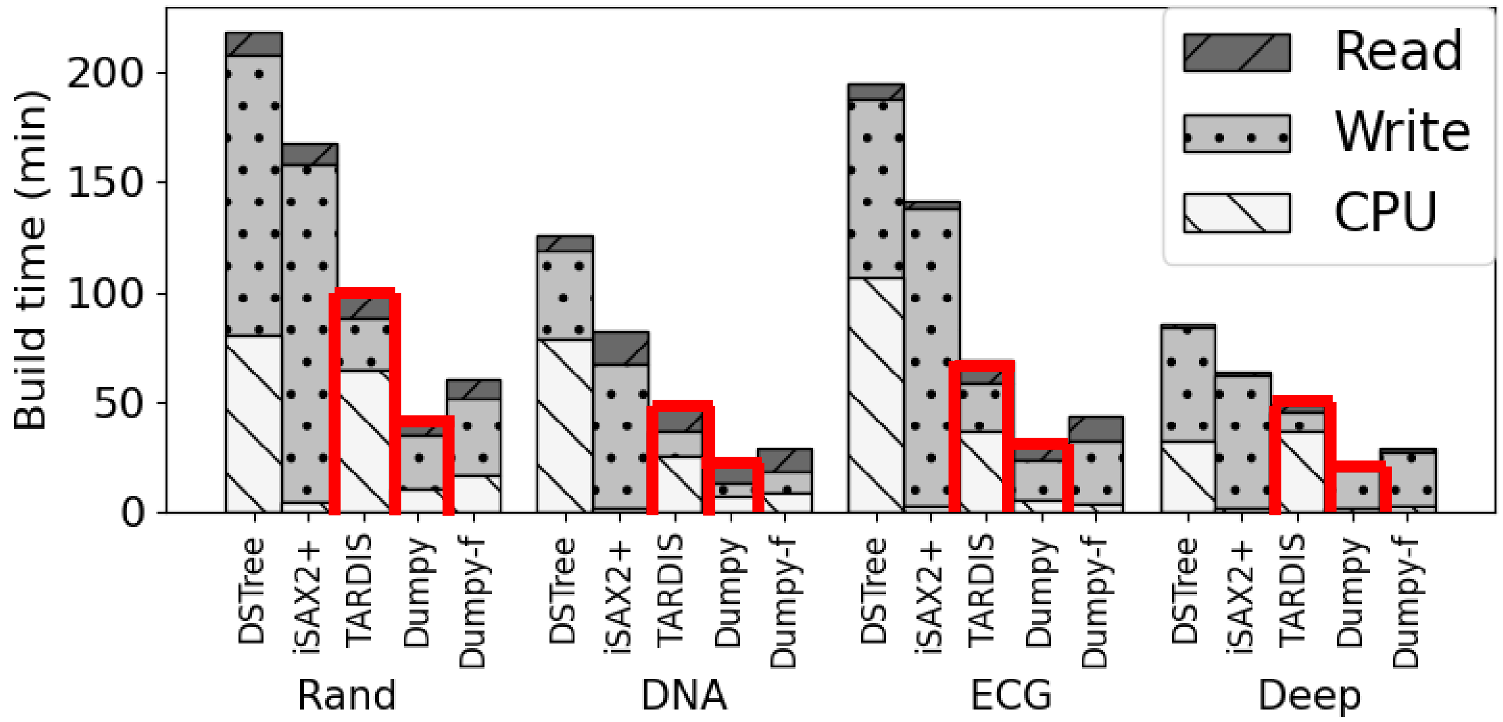
- Comparison methods
  - iSAX2+ (2-ary iSAX-based)
  - TARDIS (full-ary iSAX-based)
  - DSTree (EAPCA-based)

Dataset	***Length***	***Dataset size***
RandomWalk	256	100m (100GB)
DNA	1024	26m (113GB)
ECG	320	97m (117 GB)
Deep1B	96	100m (38GB)

13-Scalability on Index Building



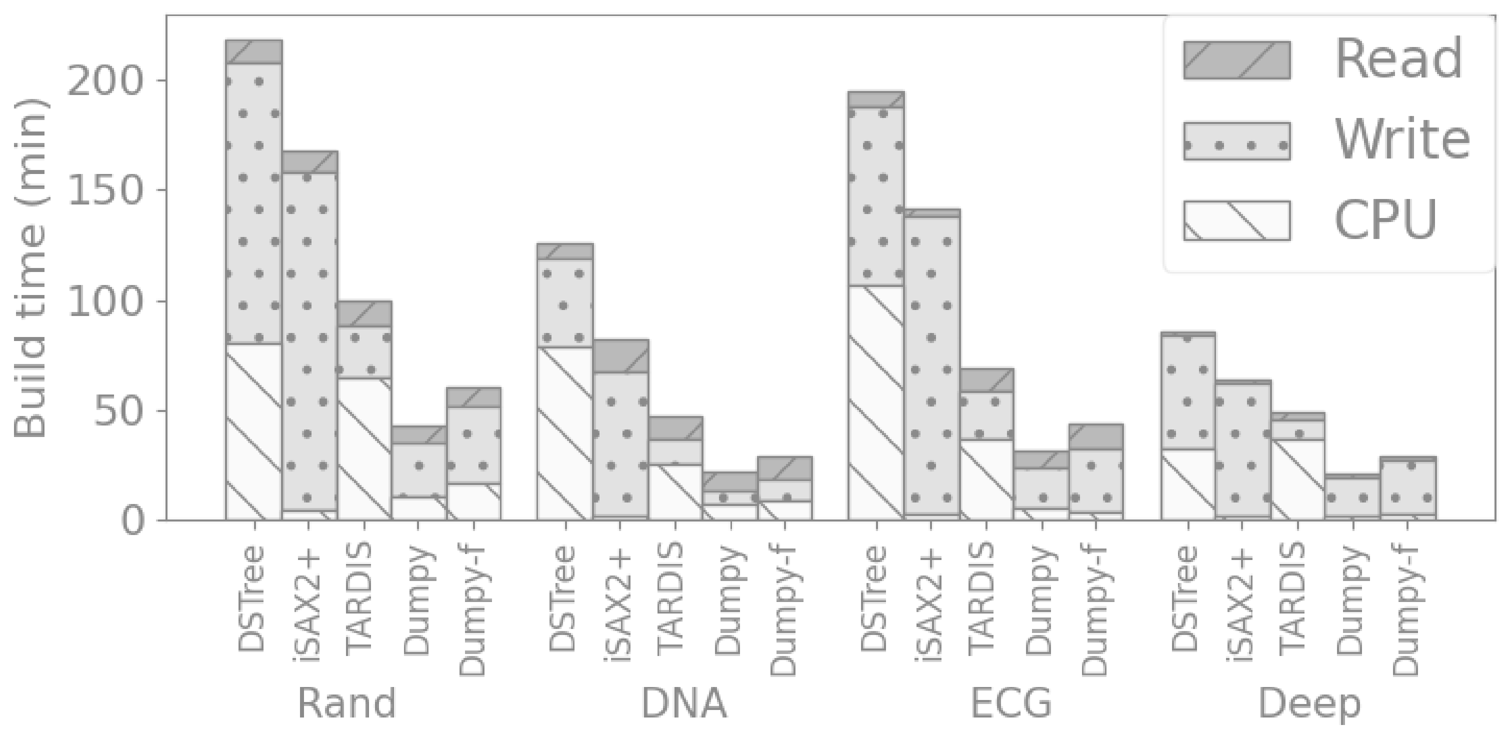
13-Scalability on Index Building

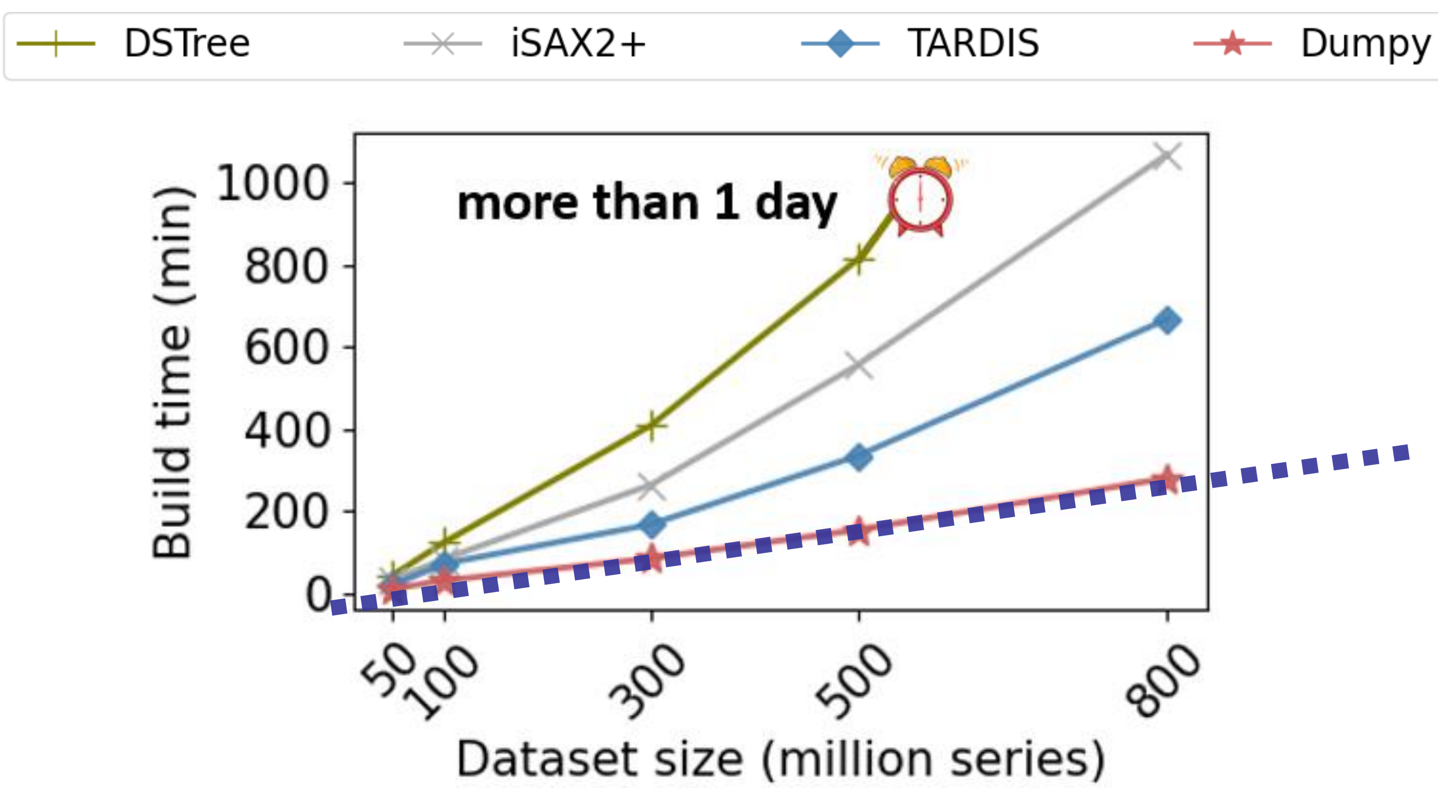


✓fastest index building, 2.5x~5.3x faster than SOTA

- Most compact structure

13-Scalability on Index Building



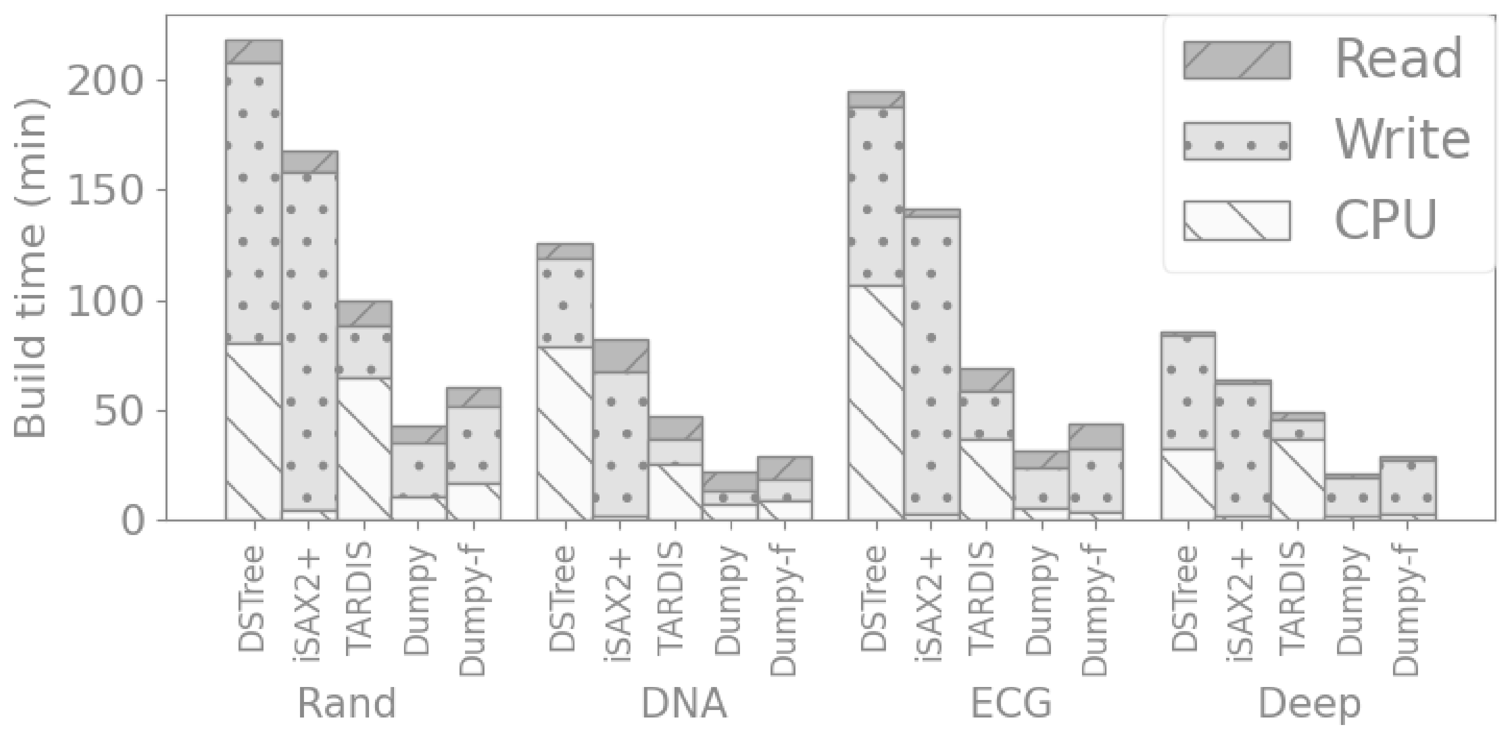


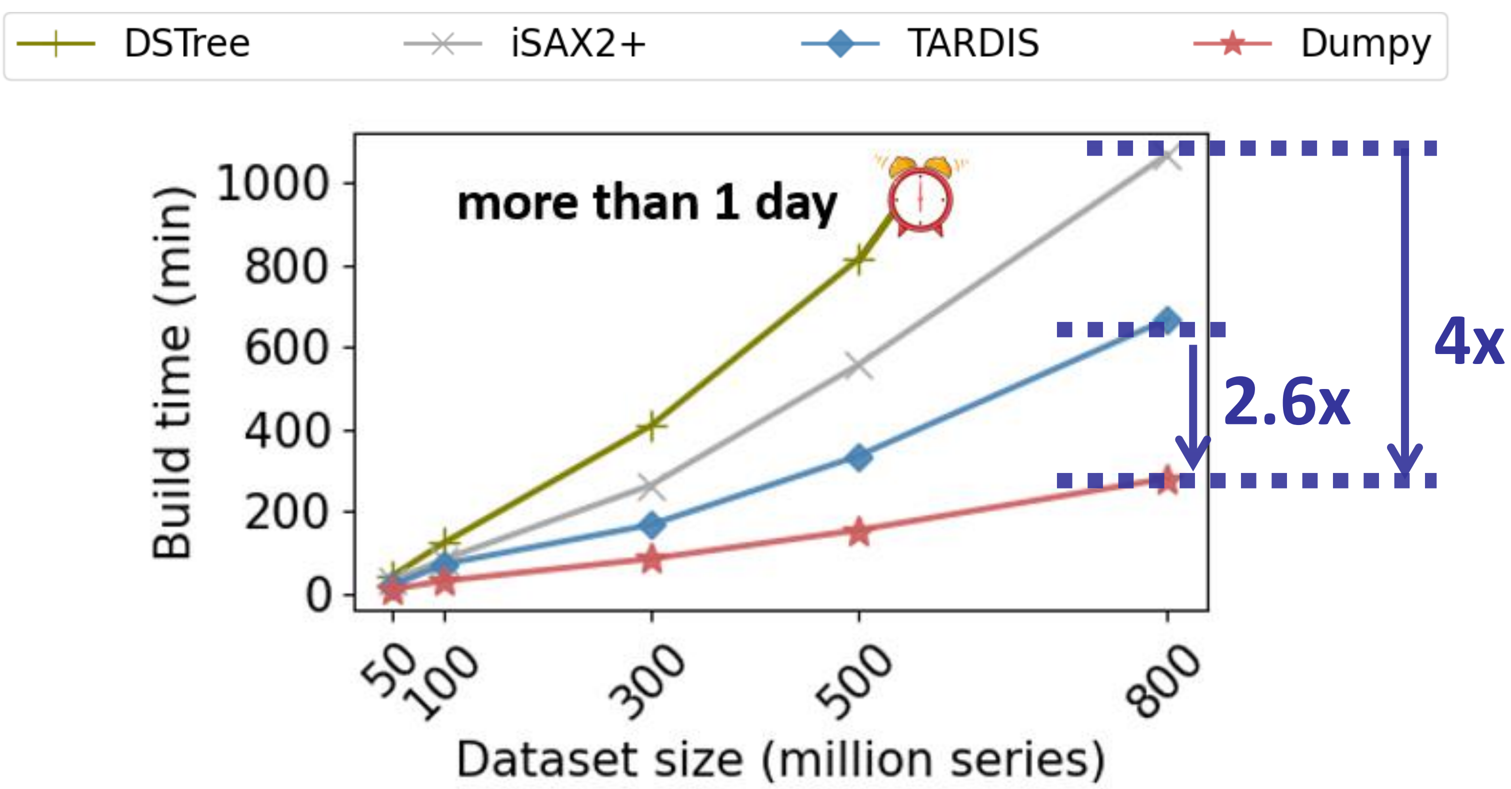
✓fastest index building, 2.5x faster than TARDIS (SOTA)

- Most compact structure

✓linear scalability (R^2> 0.99)

13-Scalability on Index Building





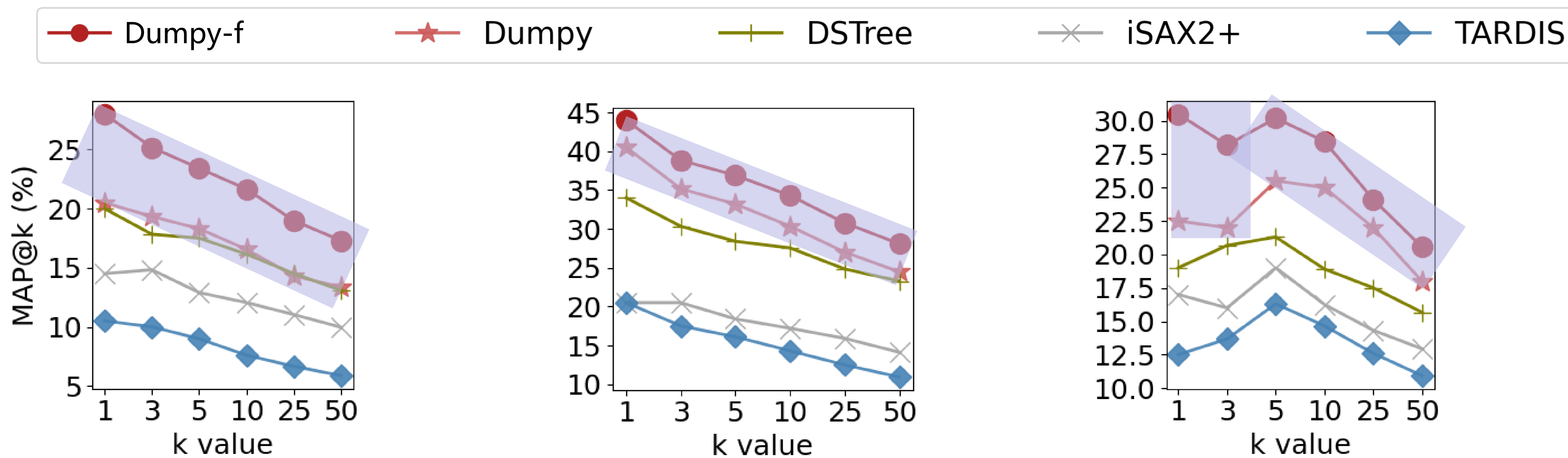
✓fastest index building, 2.5x faster than TARDIS (SOTA)

- Most compact structure

✓linear scalability

✓up to 4x faster than SOTA for 800GB data

14-Search Accuracy (search one node)

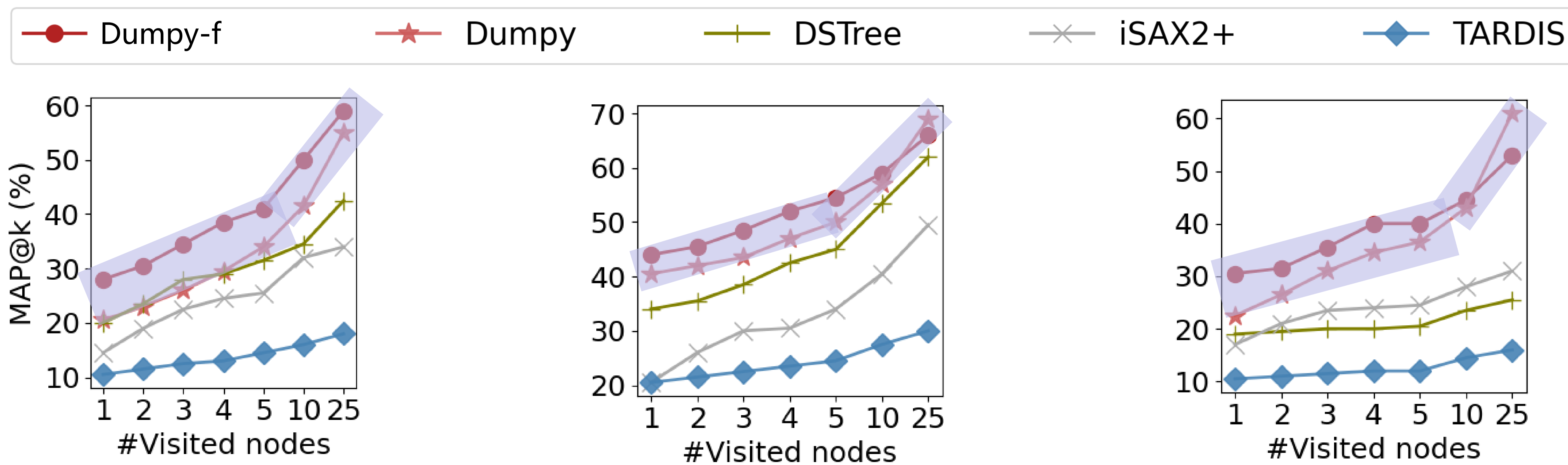


(a) Rand (b) DNA (c) ECG

✓highest MAP (Mean Average Precision)

- Best proximity of nodes
- 11%~84% higher MAP than SOTA

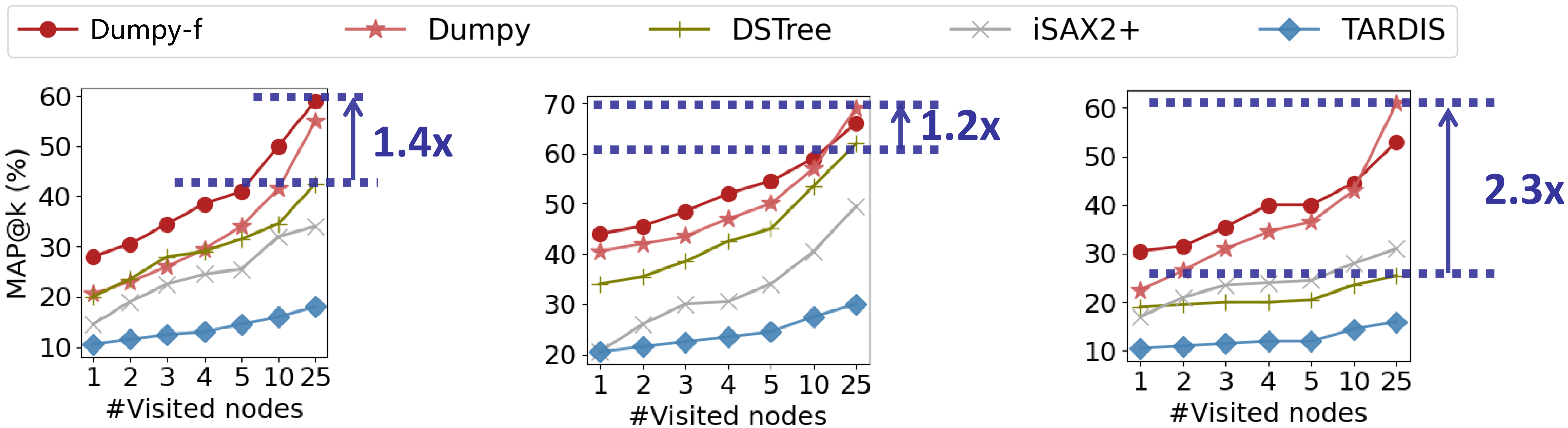
14-Search Accuracy (search more nodes)



(a) Rand (b) DNA (c) ECG

✓16%~125% higher MAP than SOTA

14-Search Accuracy (search one node)

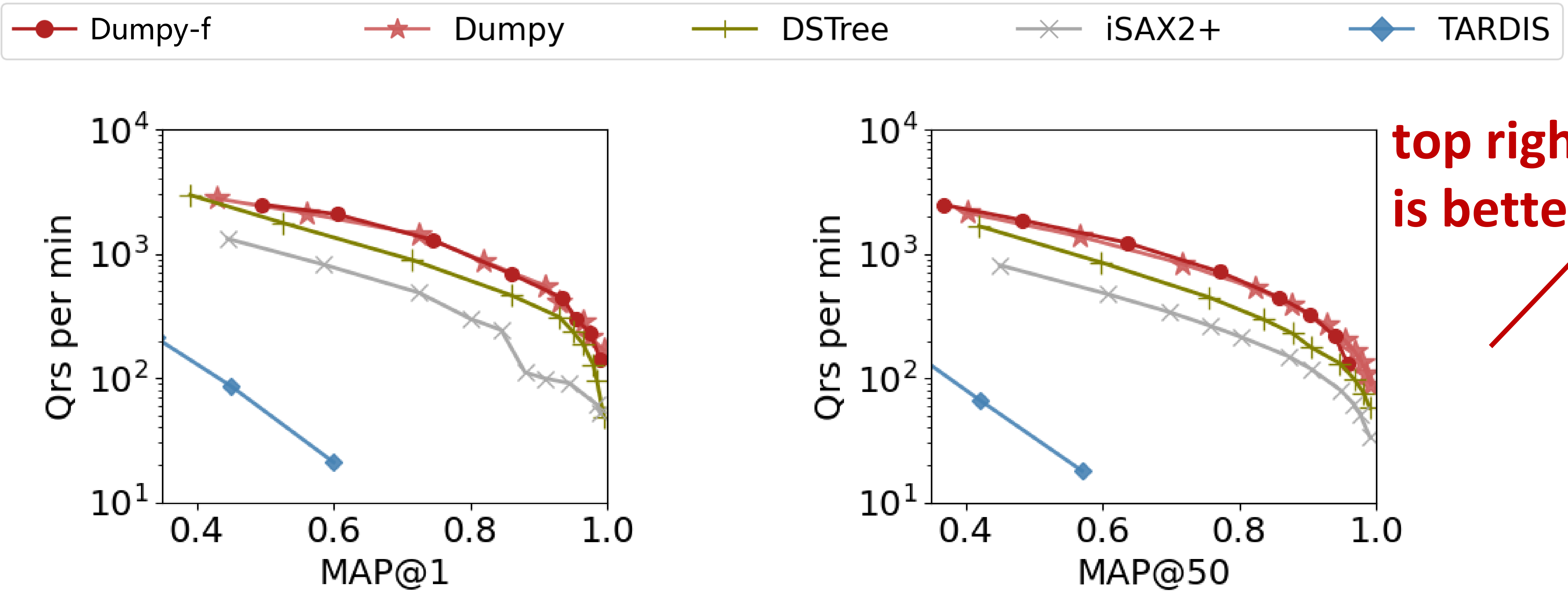


(a) Rand (b) DNA (c) ECG

✓16%~125% higher MAP than SOTA

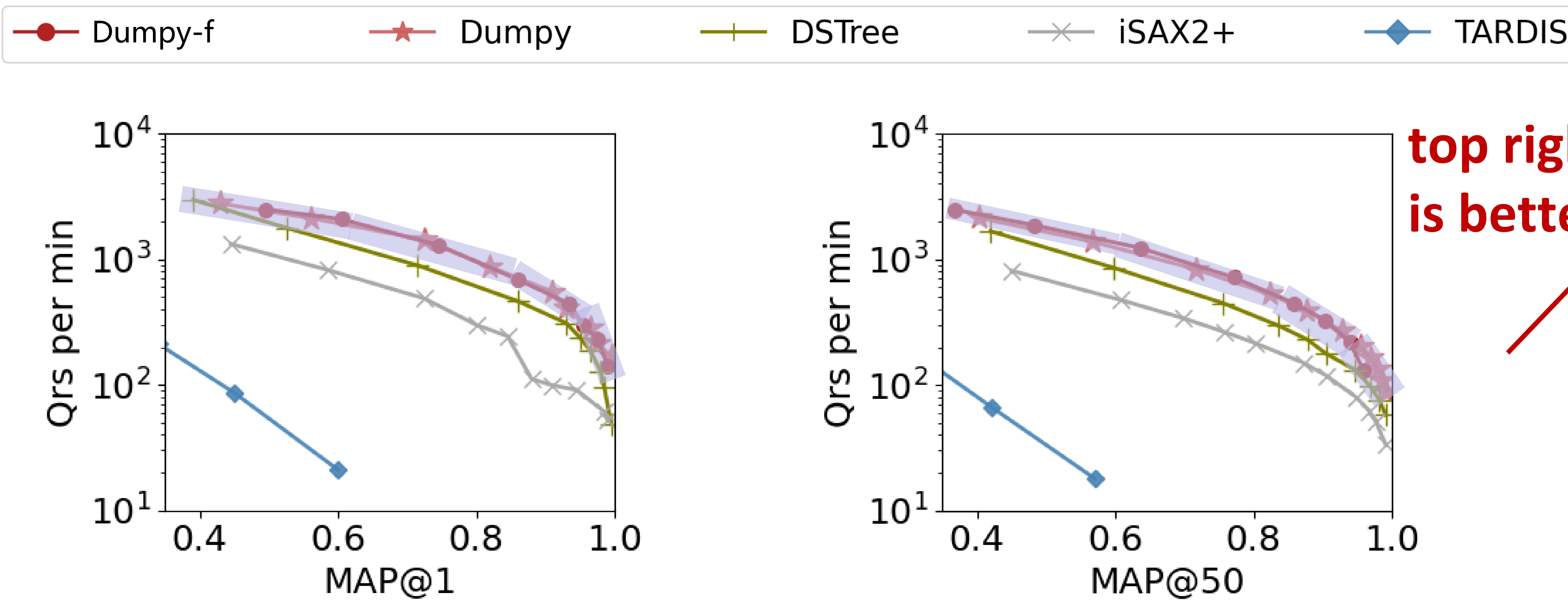
✓60% more accurate than DSTree when visiting 25 nodes (~100ms query time)

15-Pruning-based (approximate) search



top right is better

15-Pruning-based (approximate) search



top right is better

- ✓Best throughput under the same accuracy
  - ✓67% higher throughput than DSTree under 80% MAP

16-Conclusions

- Identify the inherent proximity-compactness trade-off in the structural design of SOTA iSAX-index family
- Propose Dumpy, a compact and adaptive multi-ary data series index striking the right balance of this trade-off
  - faster index-building and better scalability
  - more accurate and efficient simialrity searh
- Devise Dumpy-Fuzzy that further improves search accuracy by mitigating the hard boundary issue

Thanks!

Dumpy: A Compact and Adaptive Index for Large Data Series Collections

Zeyu Wang, Qitong Wang, Peng Wang, Themis Palpanas and Wei Wang

ACM SIGMOD 2023, Seattle

paper: https://helios2.mi.parisdescartes.fr/~themisp/publications/sigmod23-dumpy.pdf
video: https://files.atypon.com/acm/99f6febc21ad6c5a979f504caf188d9a
code: https://github.com/DSM-fudan/Dumpy