Algorithms meet Data Compression

Philip Bille

~~March 2020~~
~~May 2020~~
...
...
~~Jan 2022~~
June 2022

Plan

- Algorithms
- Data Compression
- Compressed Computation
- Research Examples
  - Random Access and Grammars (2011)
  - Top Tree Compression (2013)
  - Persistent Strings (2020)
- Applications

Algorithms

What People Think Algorithms are

9. Hvilken af nedenstående dækker bedst beskrivelsen af, hvad en algoritme er?

- ☐ En form for seksuel aktivitet
- ☐ En serie af beregningsmodeller
- ☐ En metode til måling af smerte
- ☐ En kemisk betegnelse
- ☐ En form for kirkemaleri

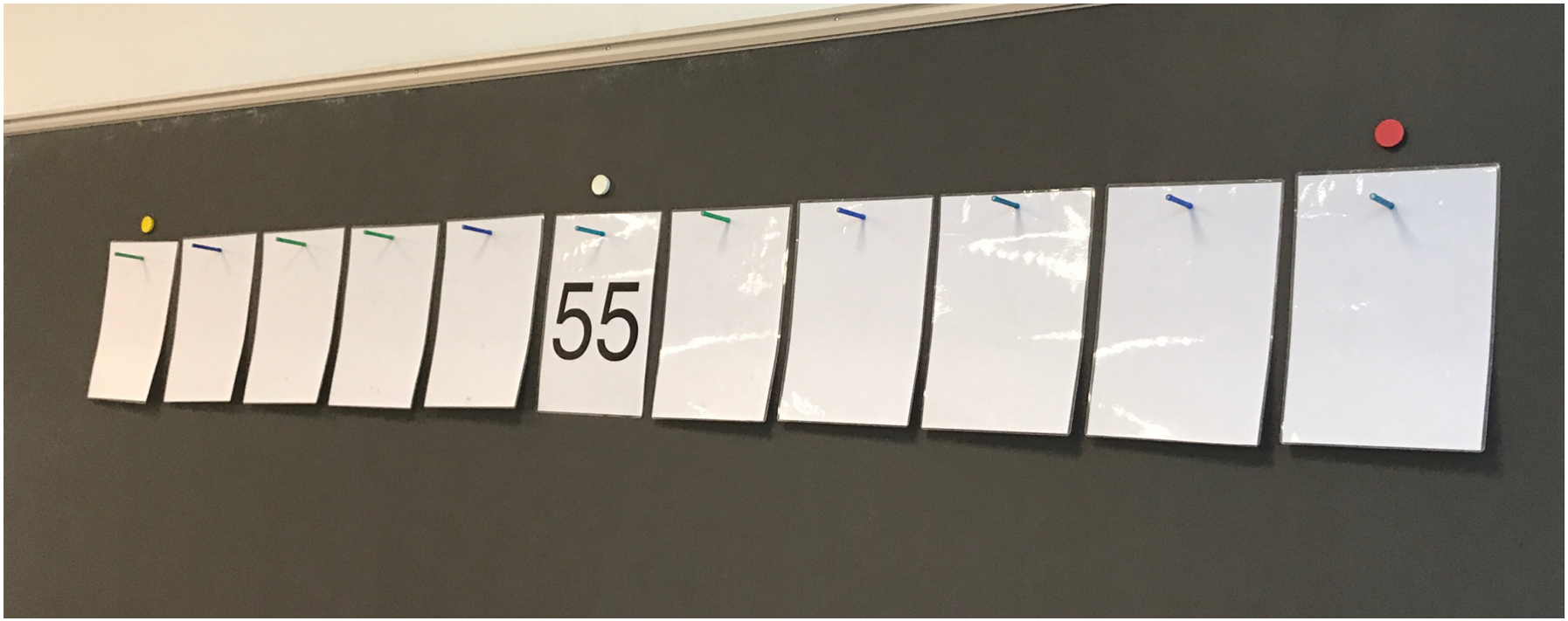
Algorithms are Processes

- Examples: Cooking from a recipe, building lego models from instructions, playing music from a sheet.
- Goals: Clarity, correctness, simplicity, performance.

- Programming is explaining an algorithm to a computer.

Searching

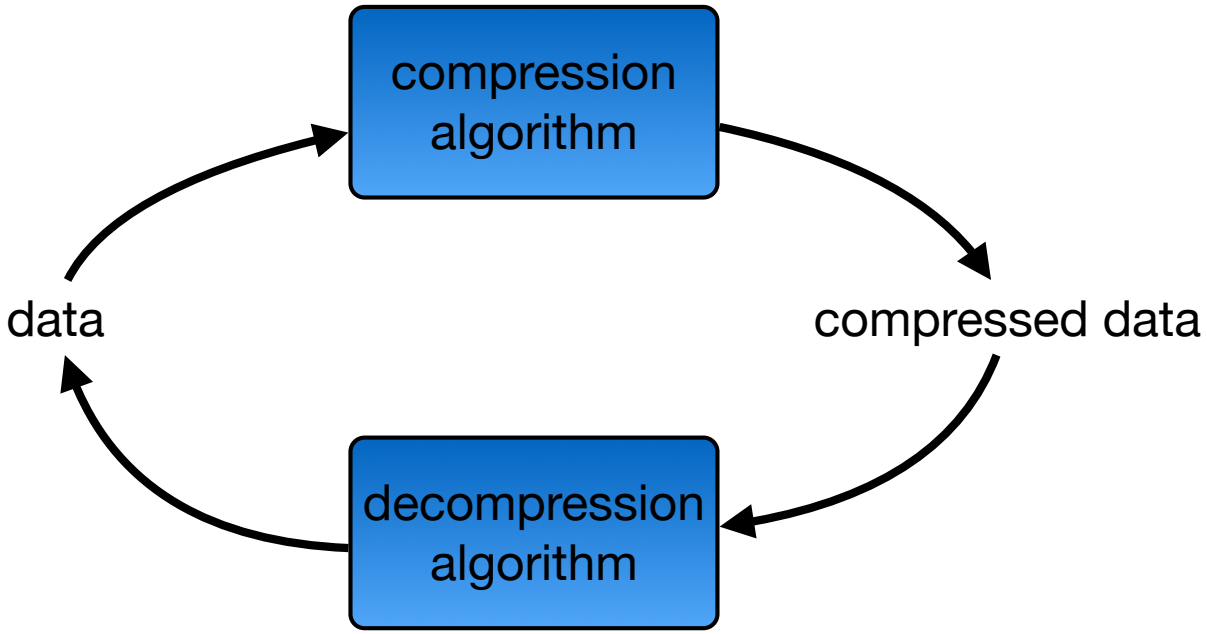
- How to find a number in a sequence of sorted numbers?



Data Compression

Data Compression

- Goal: reduce size of data.
- Lossless vs lossy compression.
- Data compression is everywhere.



Grammar Compression

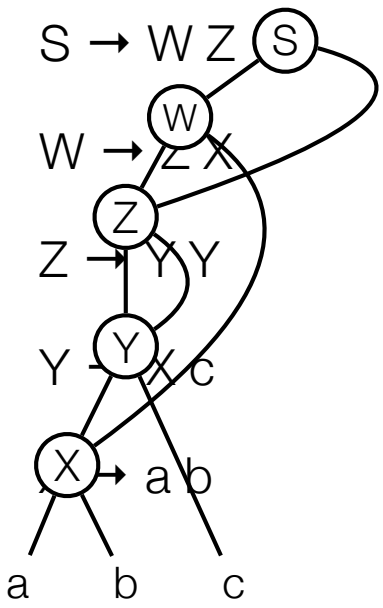
- Goal: data compression by identifying repetitions and encoding them.

a b c a b c a b a b c a b c

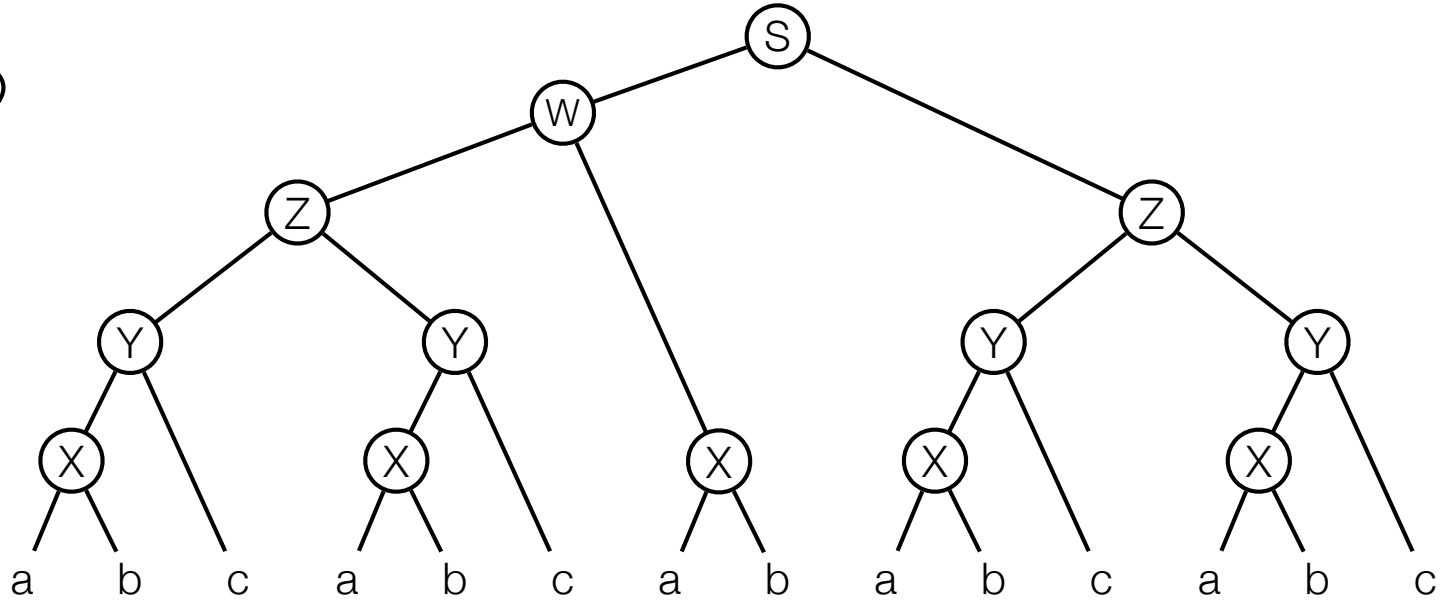
Z → a b c a b c Z a b Z

Grammar Compression

- Find a most frequent pair of consecutive symbols.
- Replace each occurrence with a new symbol.
- Add rule.
- Repeat.



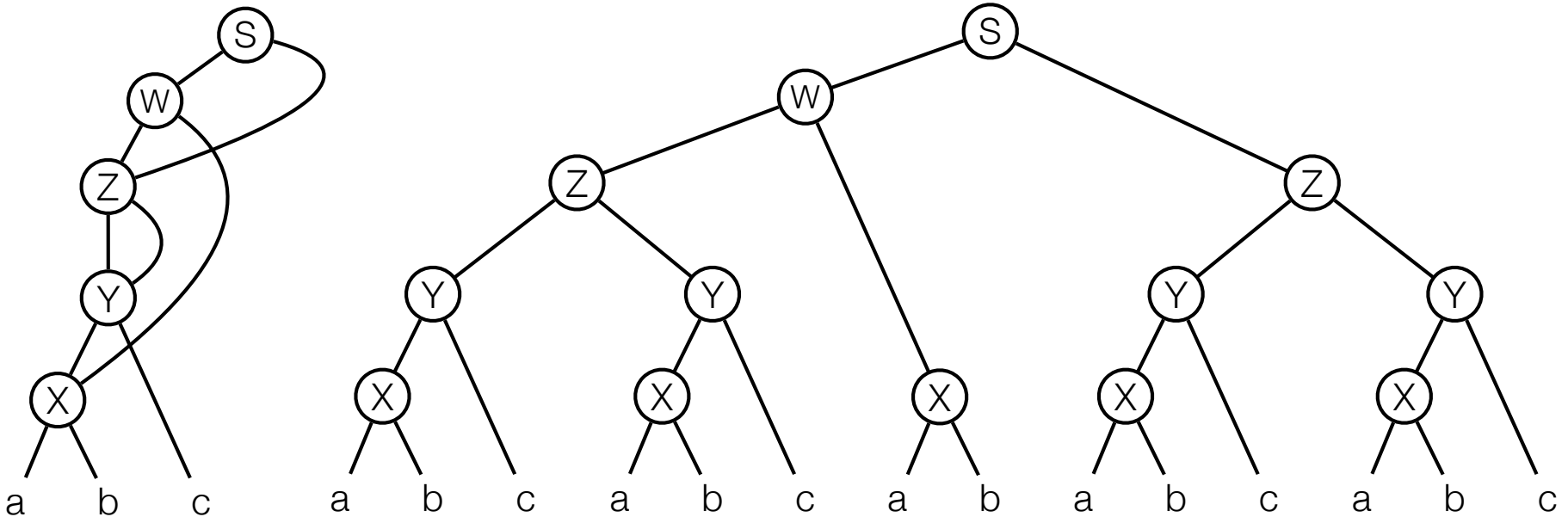
g = 5



n = 14

Grammar Decompression

- Unfold rules to get tree.



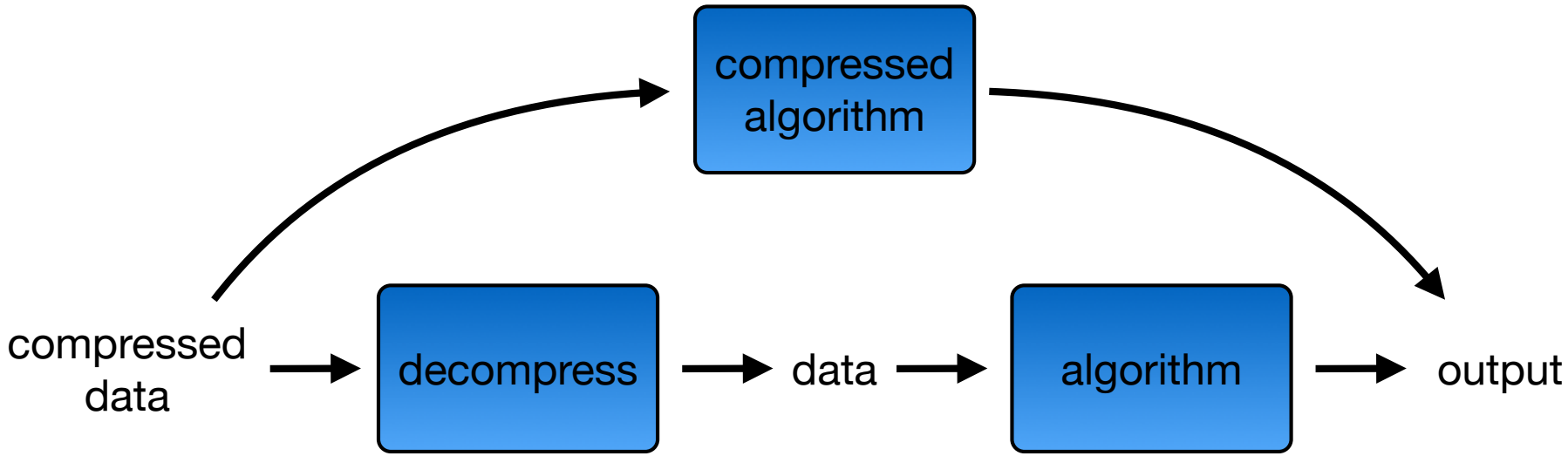
Grammar Compression

- Why grammar compression?
  - Most state-of-the-art compression schemes are essentially grammar compression schemes.
  - Simplicity.

Compressed Computation

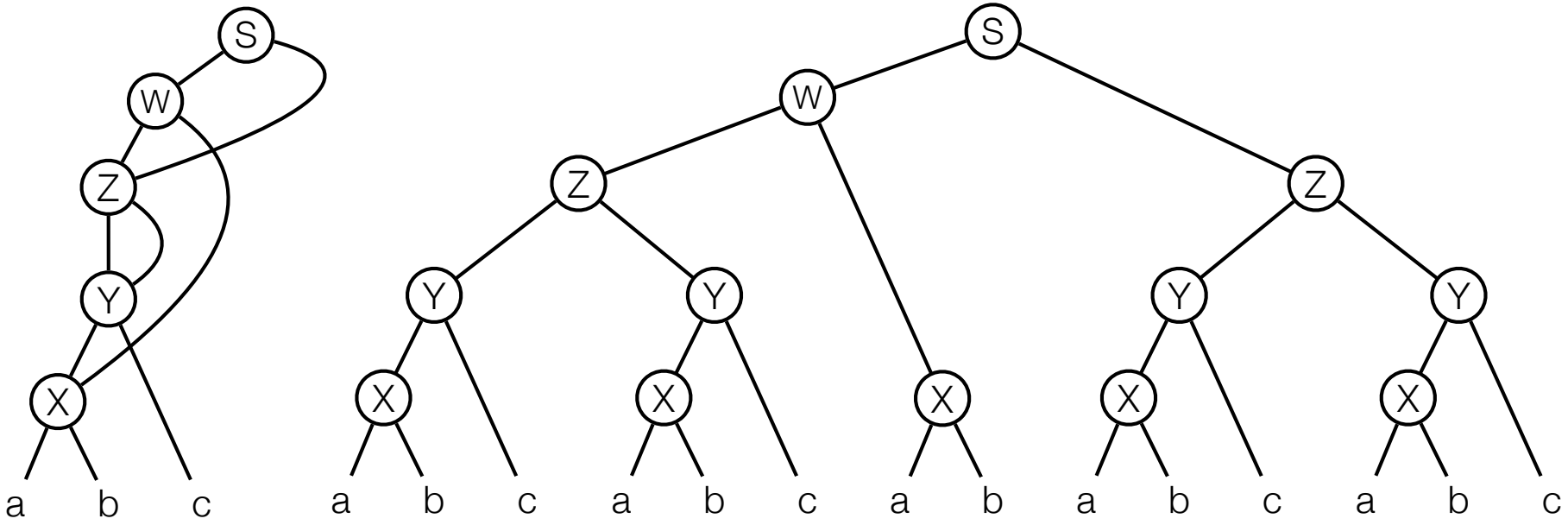
Compressed Computation

- How can we do useful computation on compressed data?



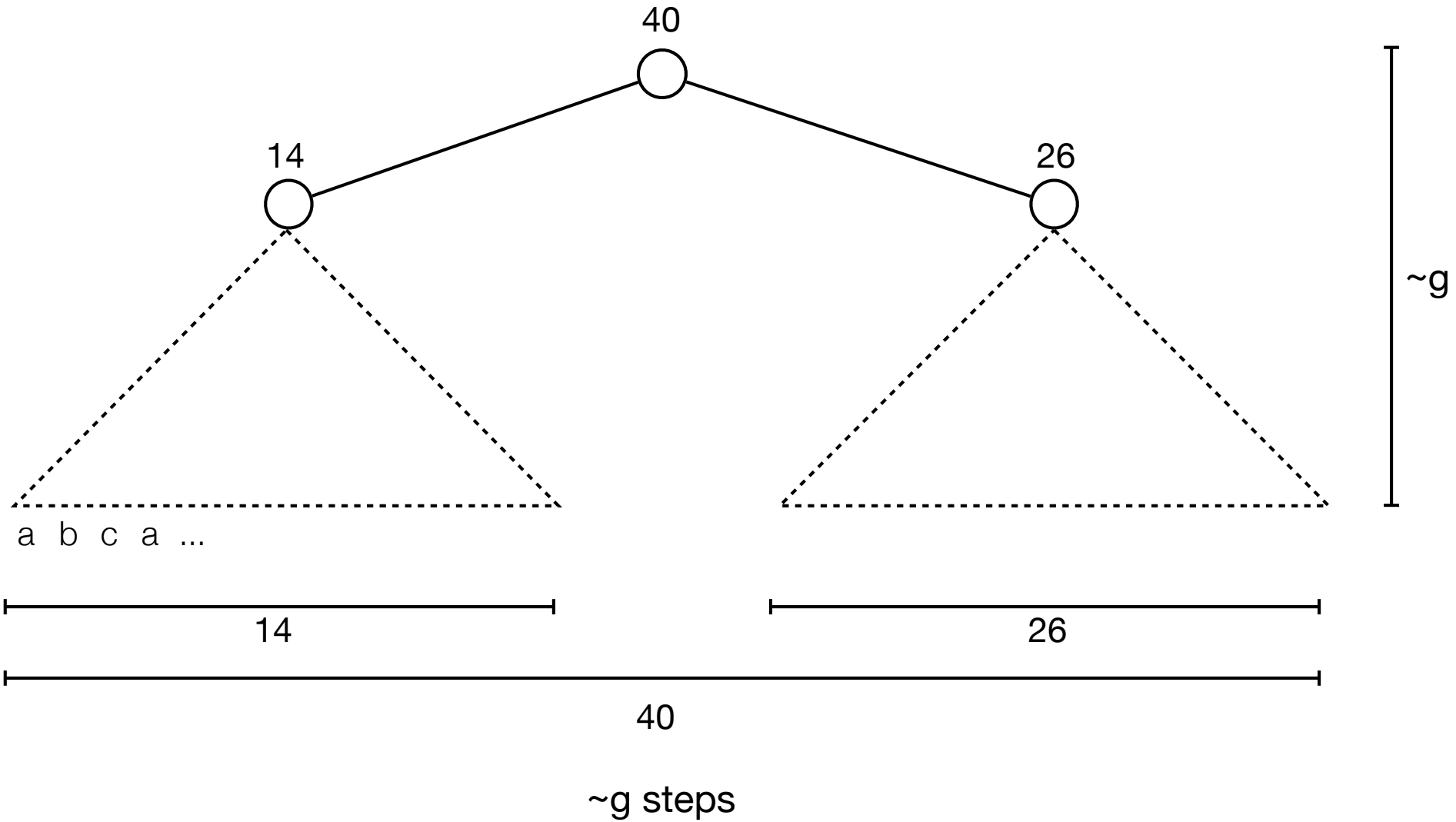
Random Access (2011)

- What is the ith character?
- What is the substring from position i to j?



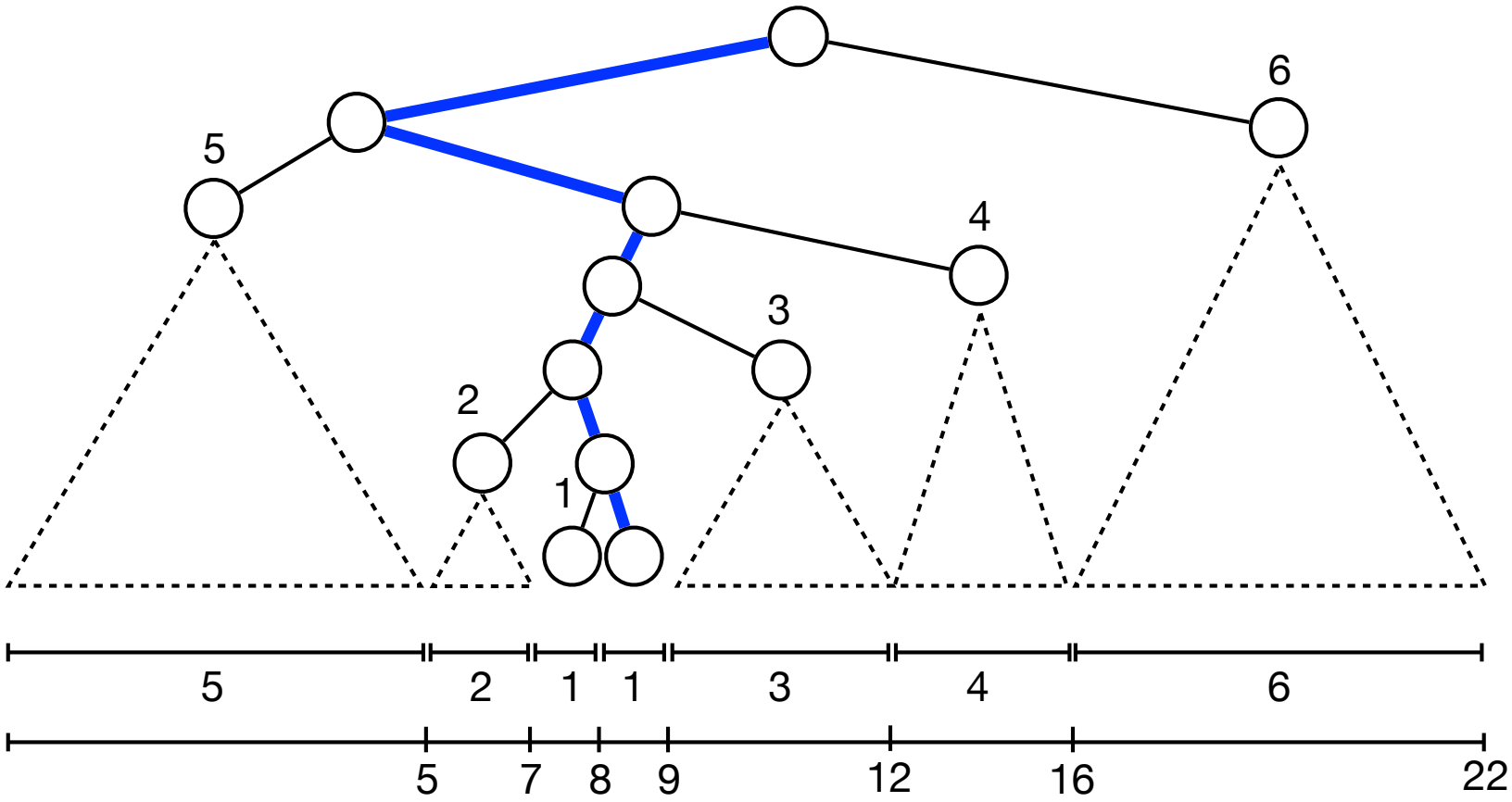
Random Access (2011)

- Store length of generated substring for each rule.
- Access: traverse grammar from top to bottom.



Random Access (2011)

- Decompose into heavy paths and store cumulative sizes of substrings along heavy paths.
- Access: traverse top-down.



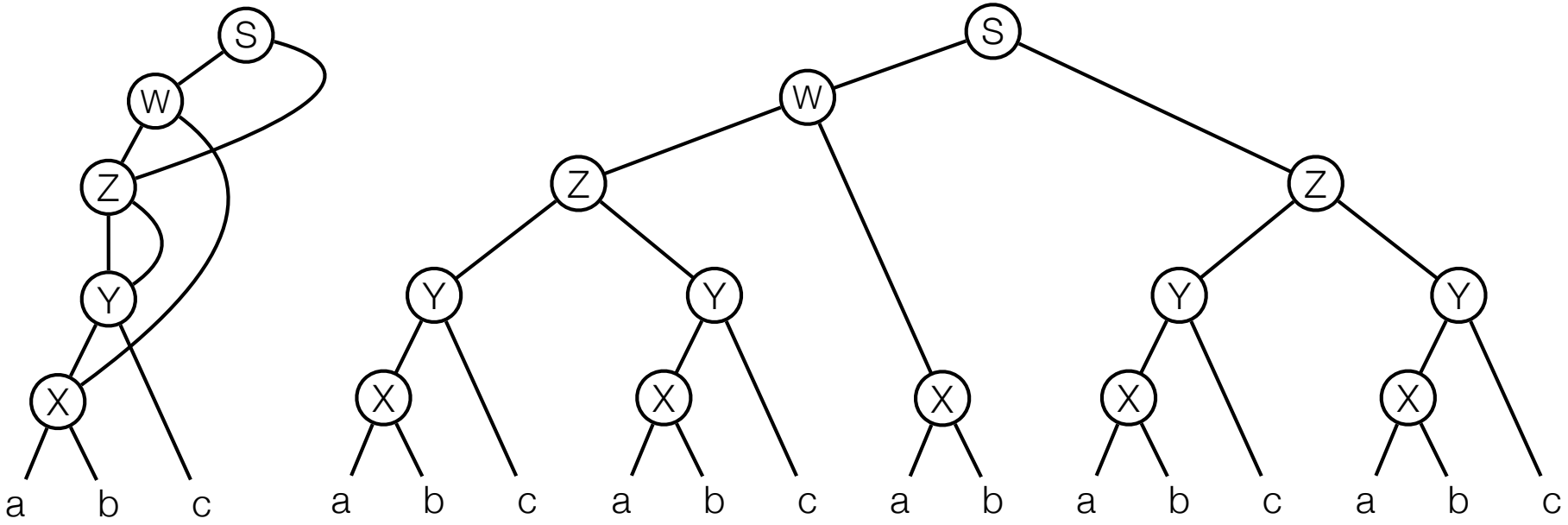
~$\log^2$ n steps optimal

Compressed Computation and Random Access

- Random access is a central component for compressed computation.
  - Compressed full-text indexing.
  - Compressed pattern matching
  - Compressed regular expression matching
  - Compressed longest common extensions.
  - Fingerprinting in compressed strings
  - Finger search in compressed strings
  - Compressed subsequence matching
  - Compressed q-gram profiling
  - ...

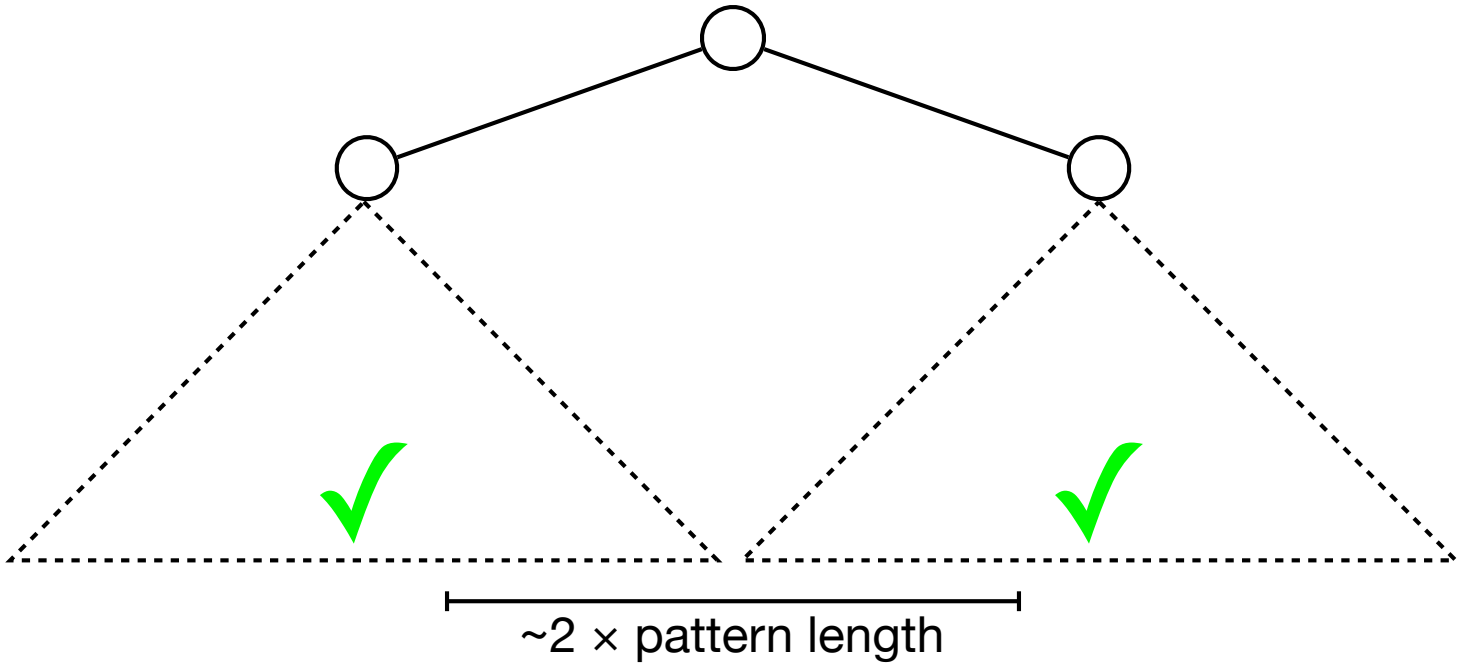
Compressed Pattern Matching (2011)

- Does pattern "cab" appear in string?



Compressed Pattern Matching (2011)

- Check each rule bottom-up.



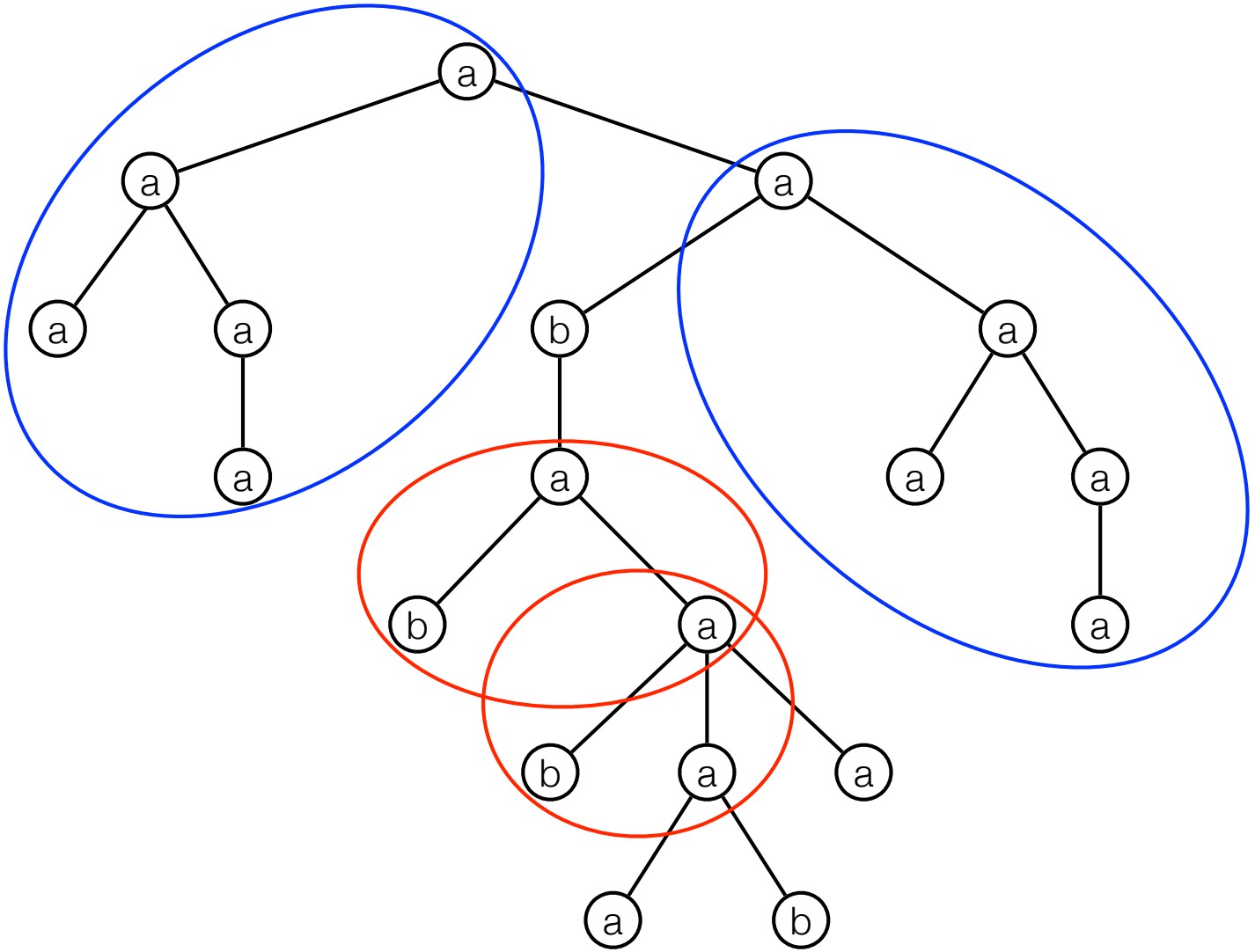
Structured Data

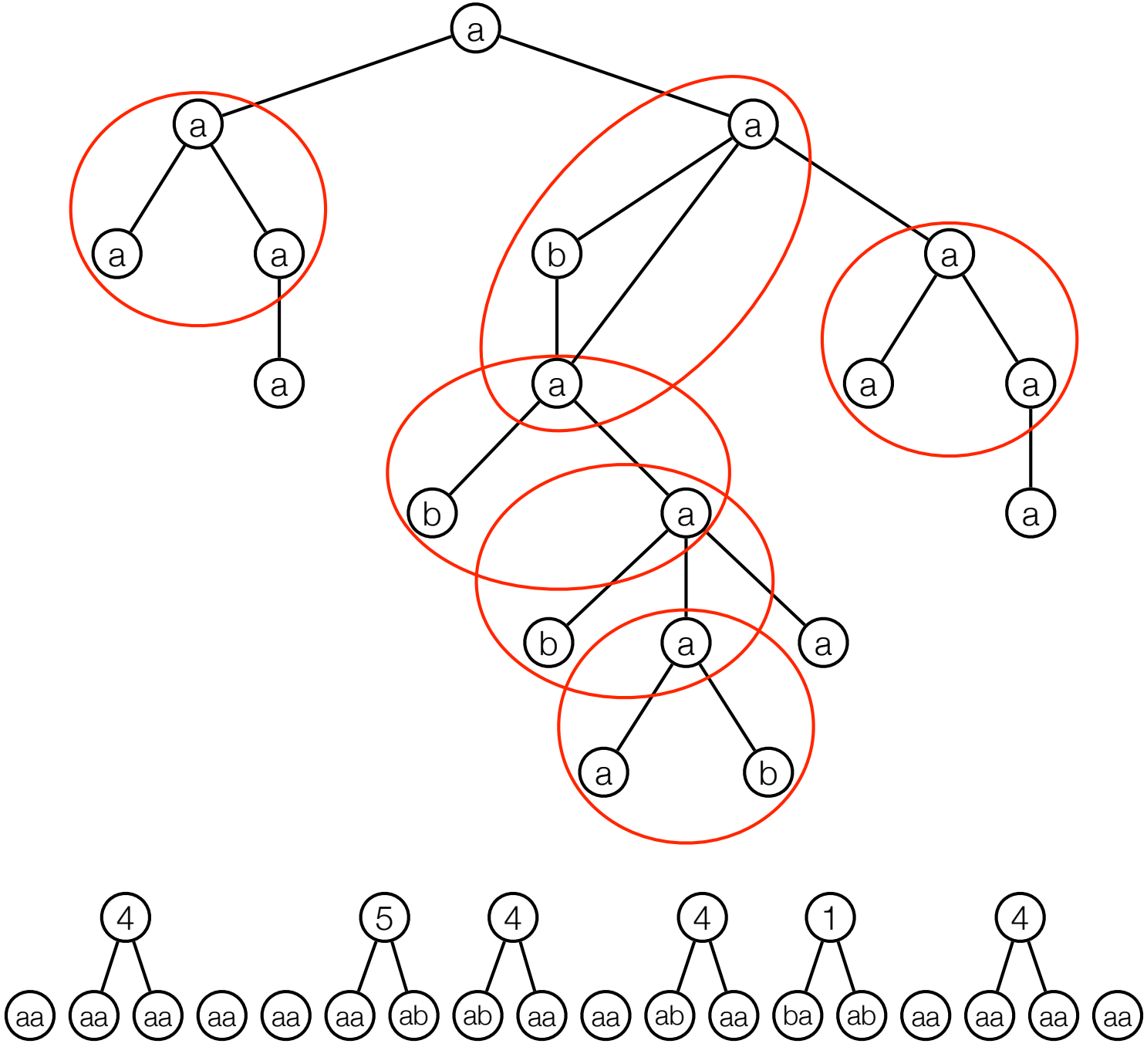
Structured Data

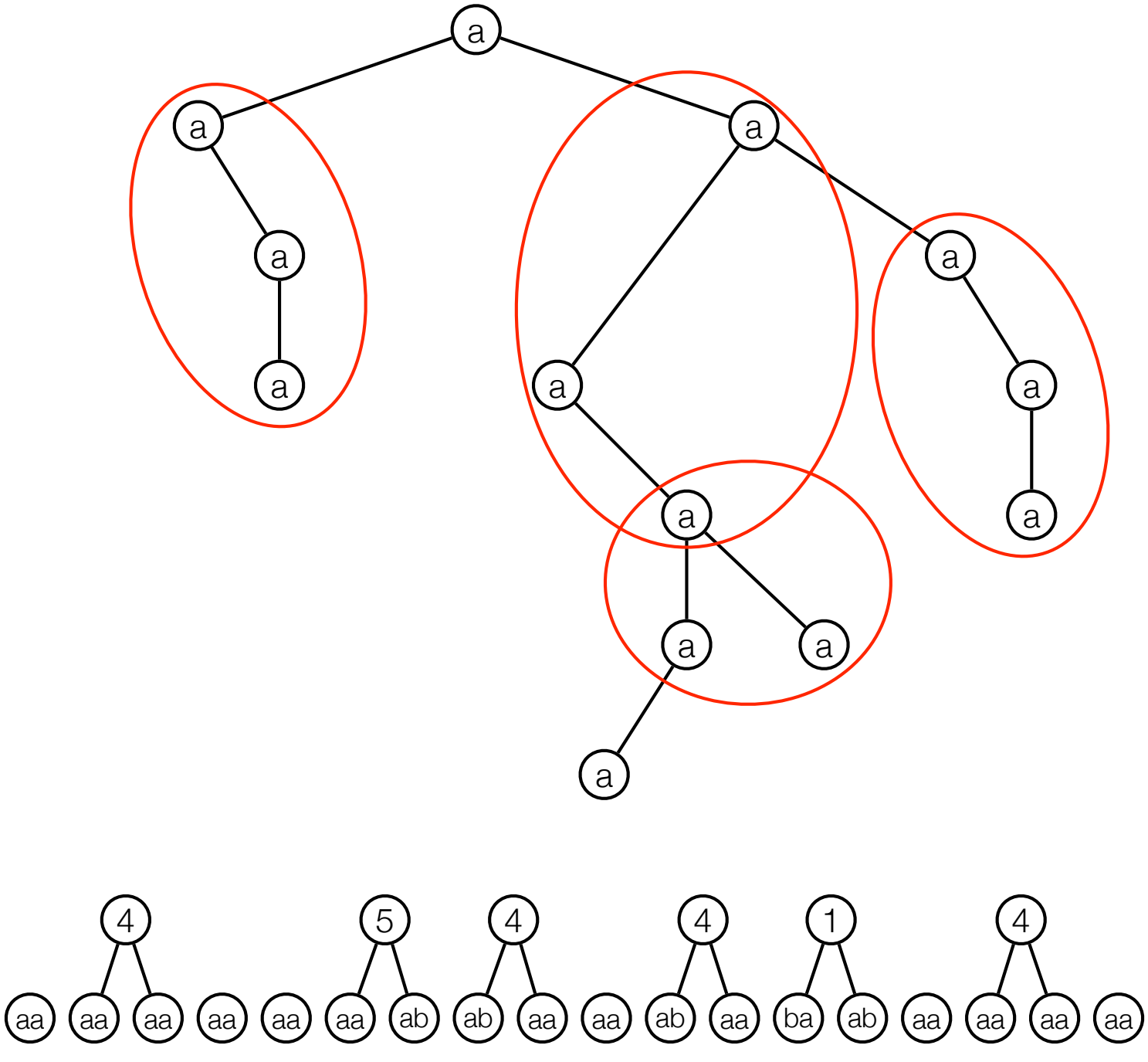
- Data can have structure: networks, hierarchical data, geographic data, relational data, images, video, ...
- Goal: Compress structure and support efficient compressed computation on structure.

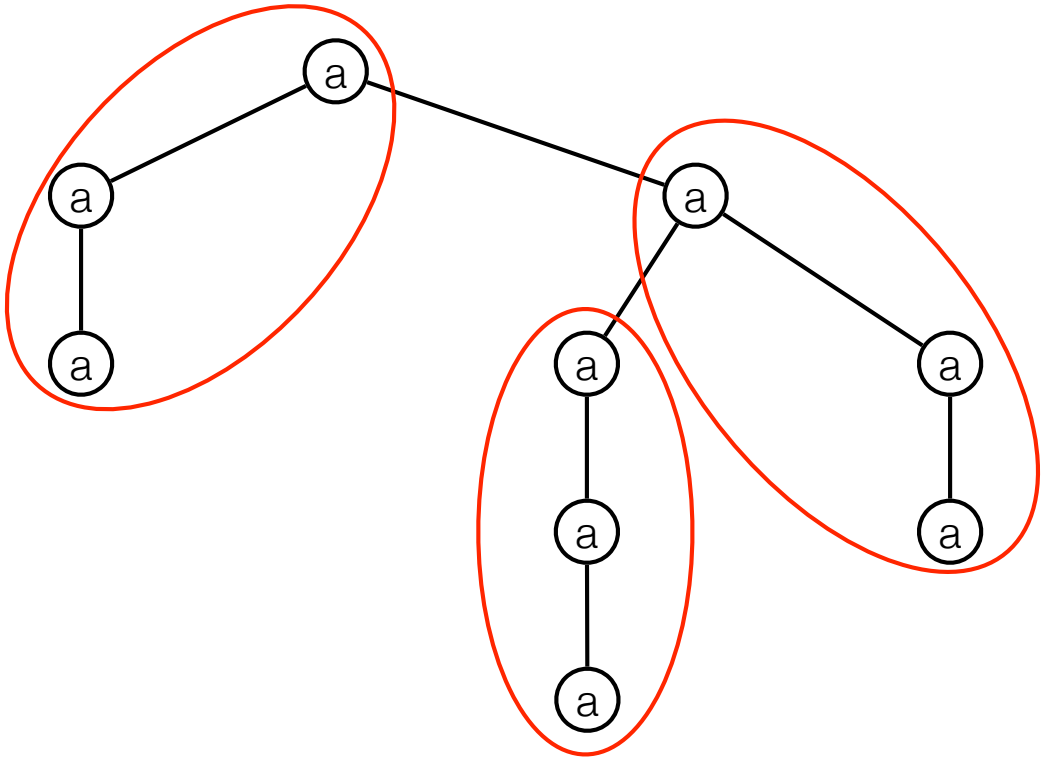
Top Tree Compression (2013)

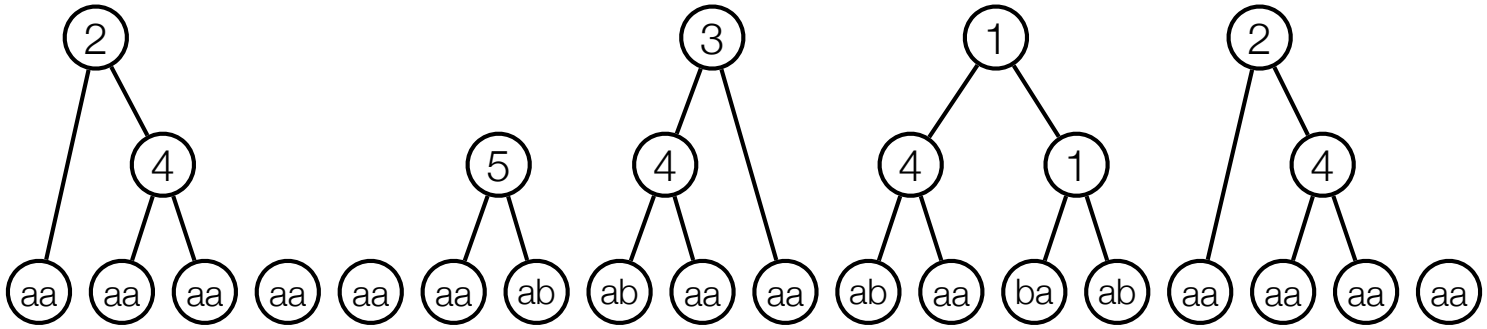
- Goal: Identify internal repeated patterns in tree and compress them.

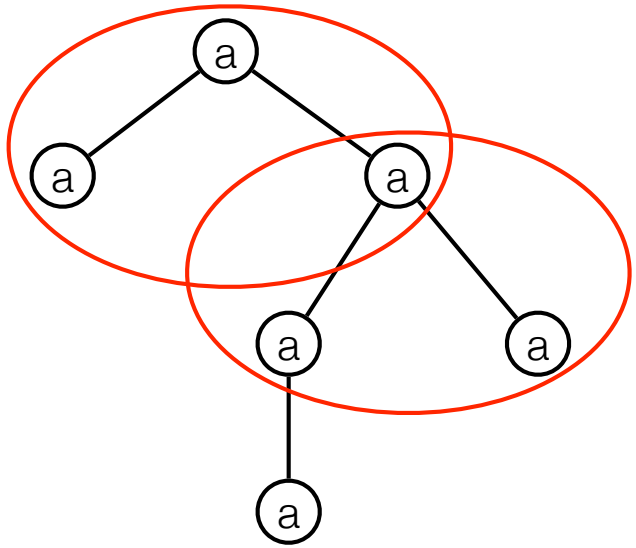


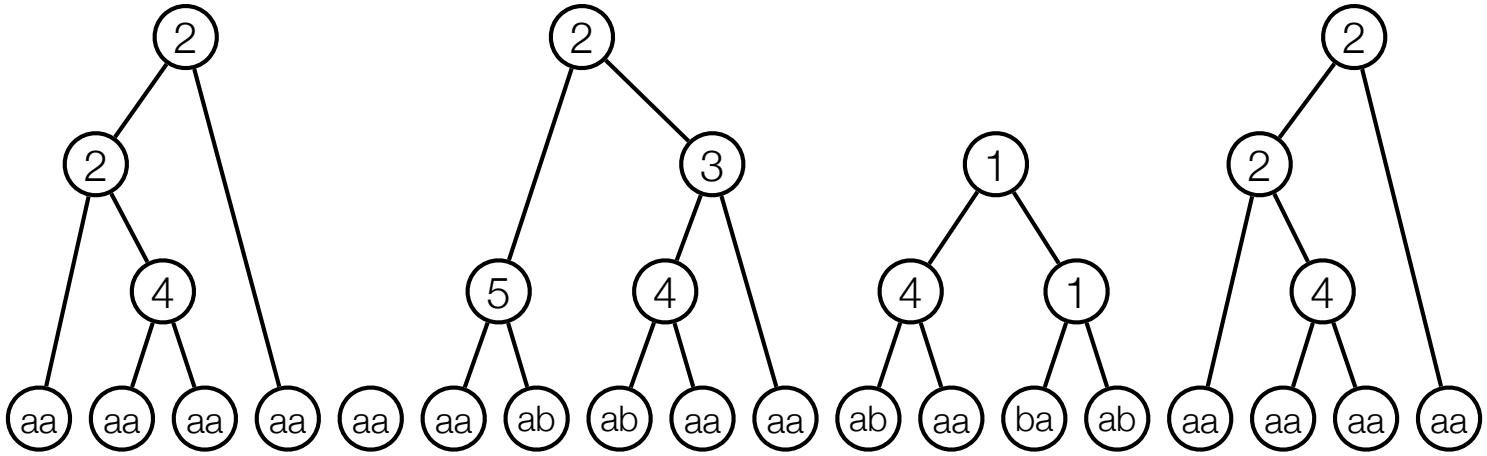

a
a
a
a
a
a
b
a
a
a
b
a
a
b
a
a
a
b
4
5
4
4
1
4
aa
aa
aa
aa
aa
aa
ab
ab
aa
aa
ab
aa
ba
ab
aa
aa
aa
aa

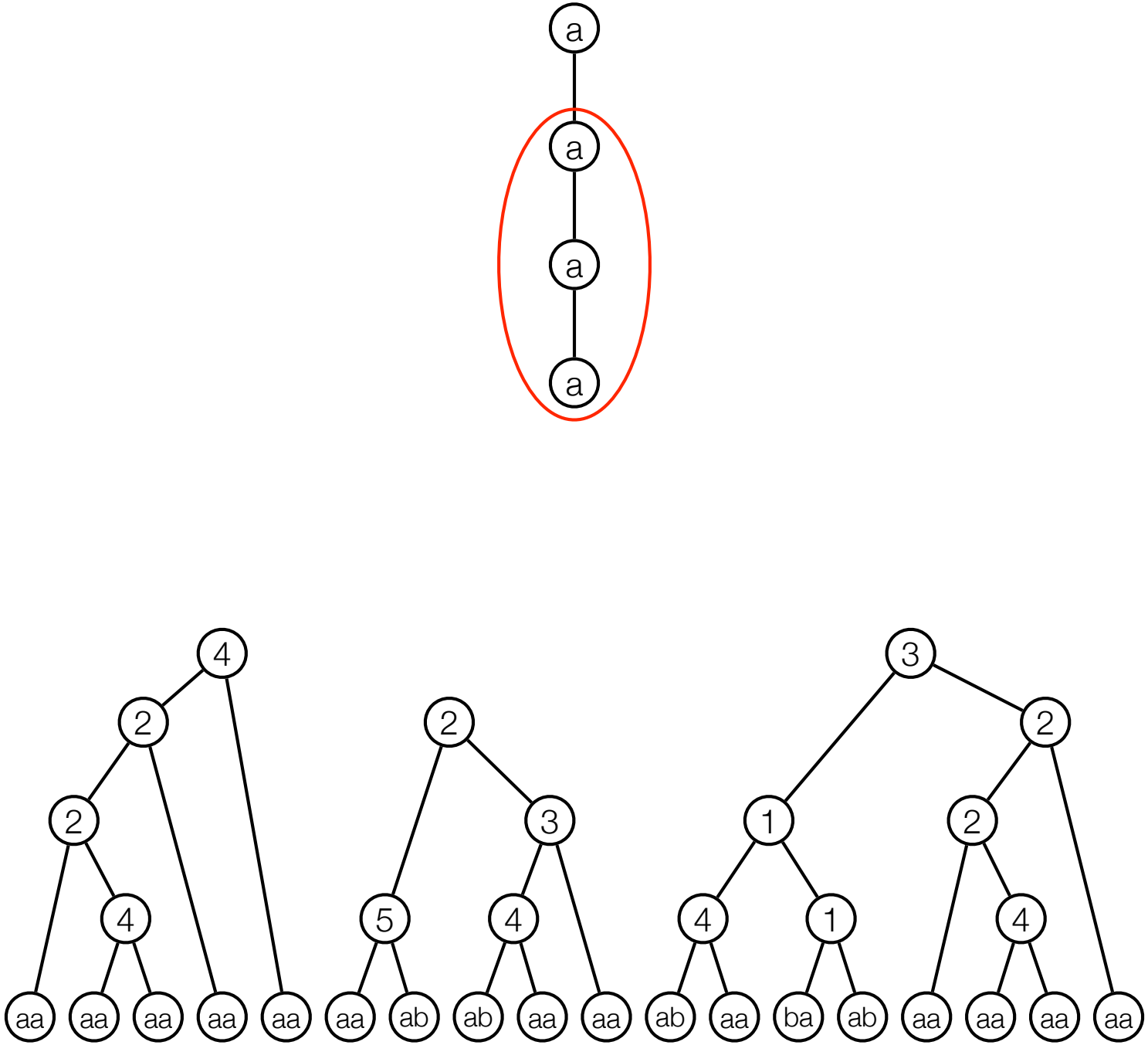

a
a
a
a
a
a
a
a
a
a
a
a
a
4
5
4
4
1
4
aa
aa
aa
aa
aa
aa
ab
ab
aa
aa
ab
aa
ba
ab
aa
aa
aa
aa

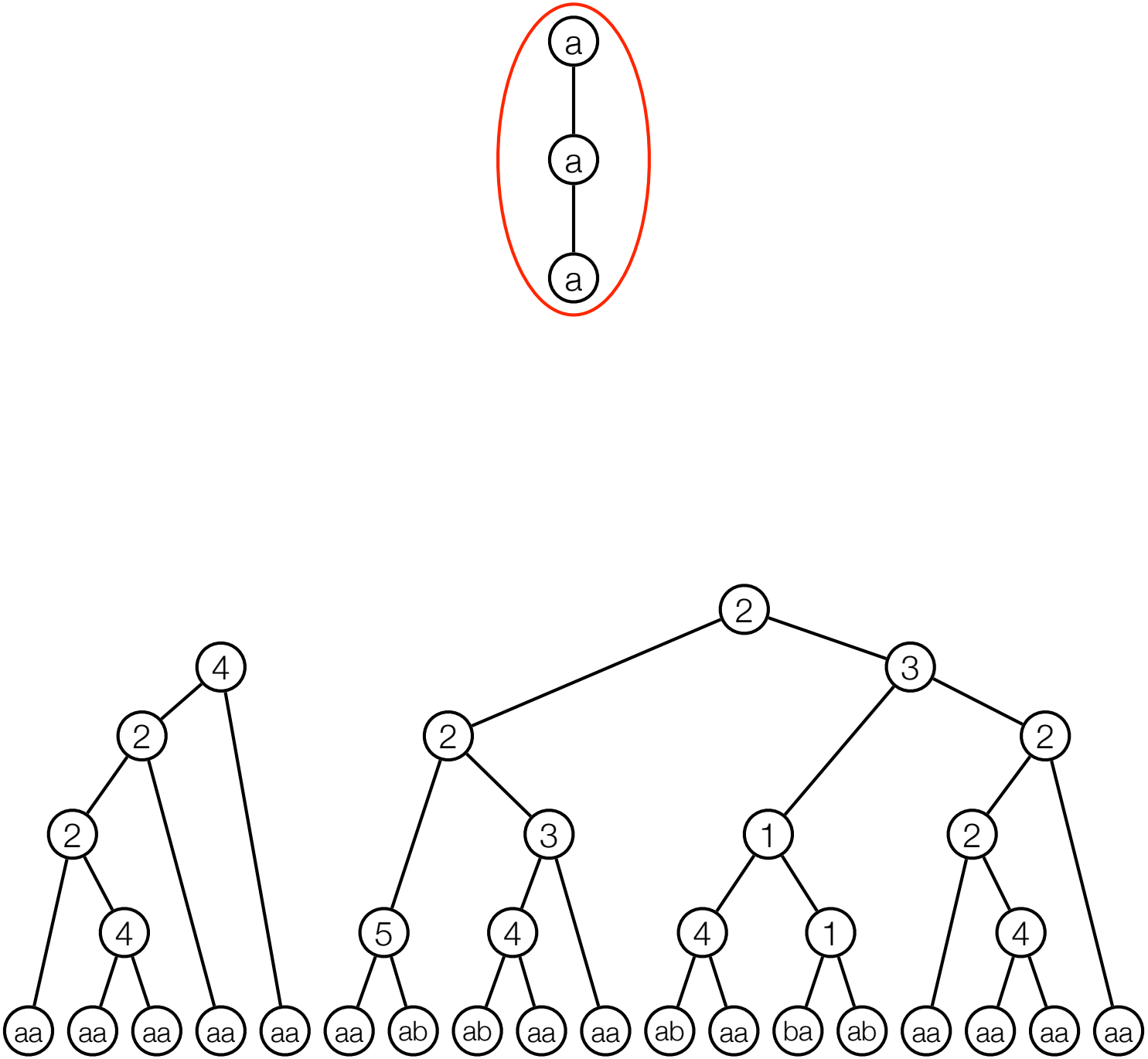

a
a
a
a
a
a
a
a

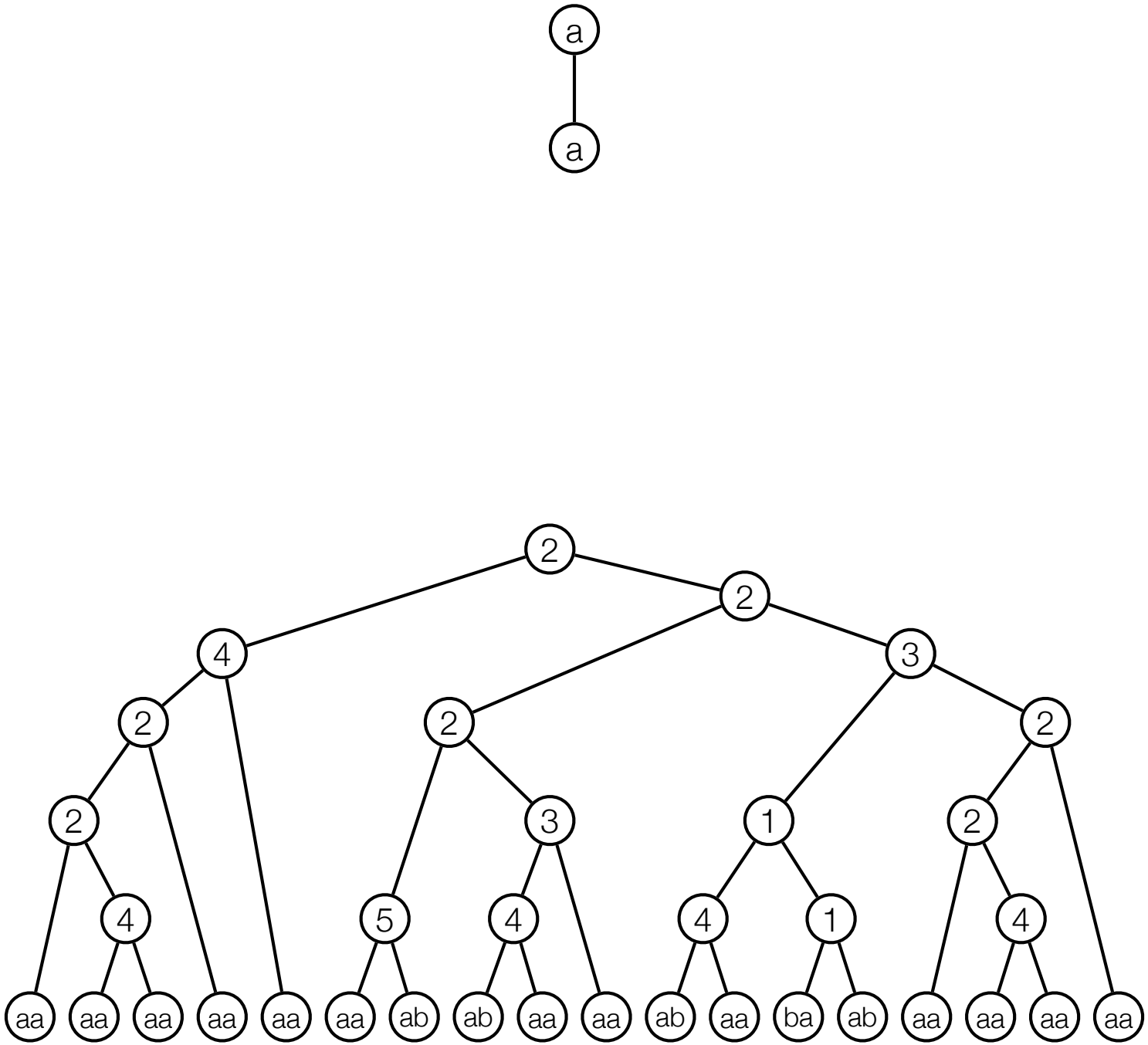

2
4
aa
aa
aa
aa
aa
5
aa
ab
3
4
ab
aa
aa
1
4
1
ab
aa
ba
ab
2
4
aa
aa
aa
aa

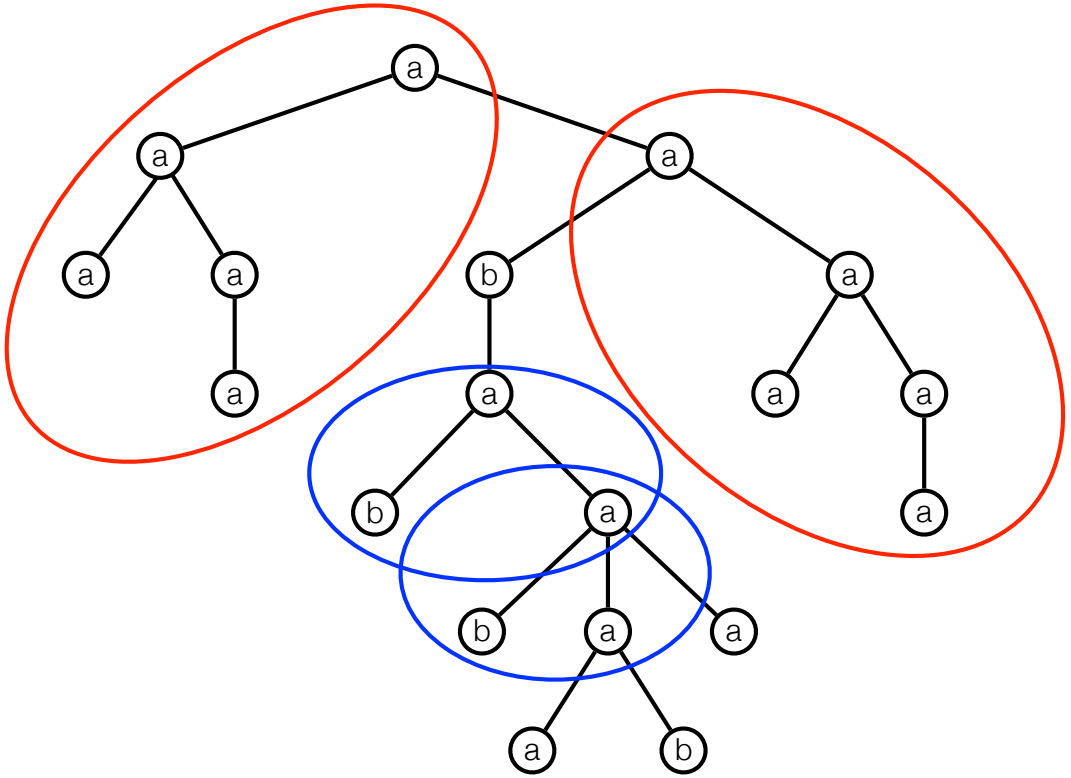

a
a
a
a
a
a

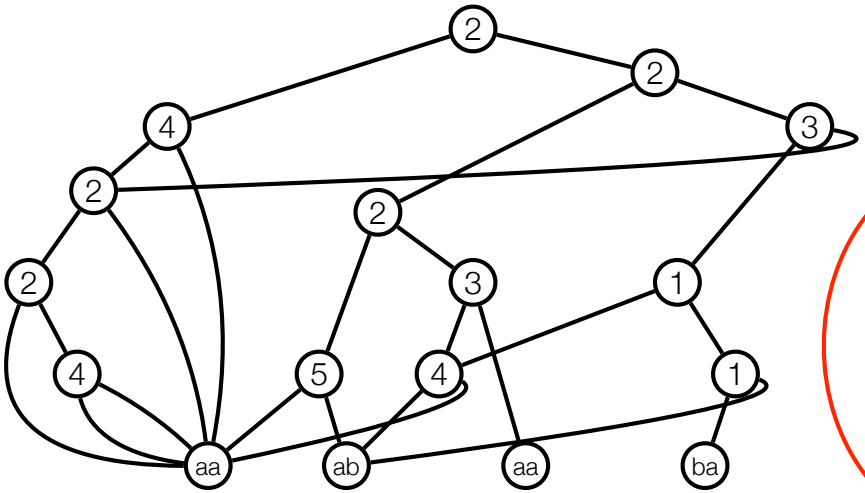

2
2
4
aa
aa
aa
aa
aa
2
5
3
4
aa
ab
ab
aa
aa
1
4
1
ab
aa
ba
ab
2
2
4
aa
aa
aa
aa

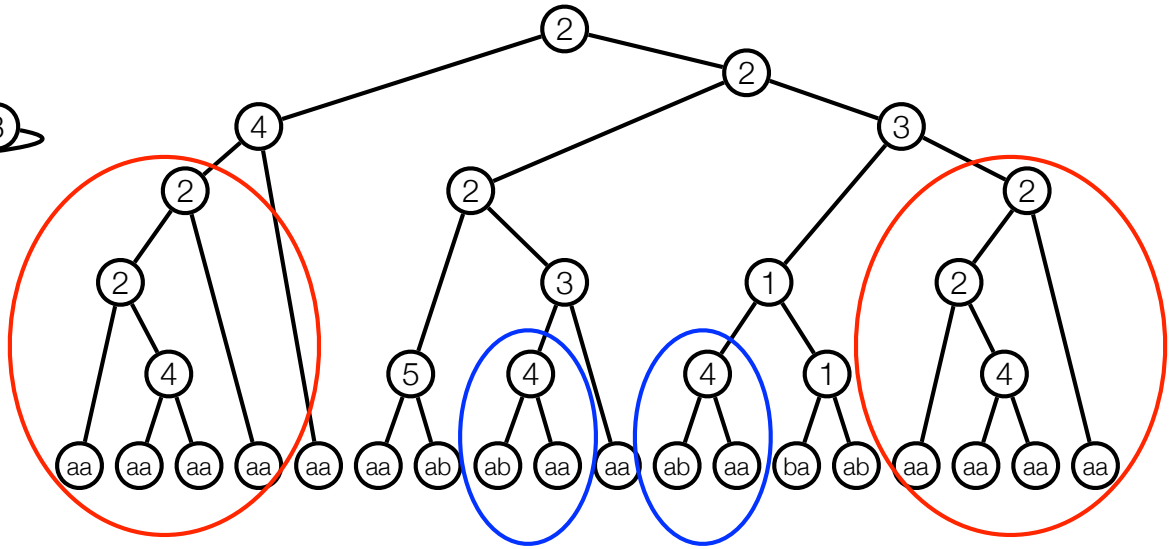

a
a
a
a
4
2
2
4
aa
aa
aa
aa
aa
2
3
5
4
aa
ab
ab
aa
aa
3
2
1
2
4
1
4
ab
aa
ba
ab
aa
aa
aa
aa


a
a
a
4
2
2
4
aa
aa
aa
aa
aa
2
2
3
5
4
aa
ab
ab
aa
aa
2
3
1
4
1
ab
aa
ba
ab
2
2
4
aa
aa
aa
aa


a
a
2
2
4
3
2
2
2
2
3
1
2
4
5
4
4
1
4
aa
aa
aa
aa
aa
aa
ab
ab
aa
aa
ab
aa
ba
ab
aa
aa
aa
aa

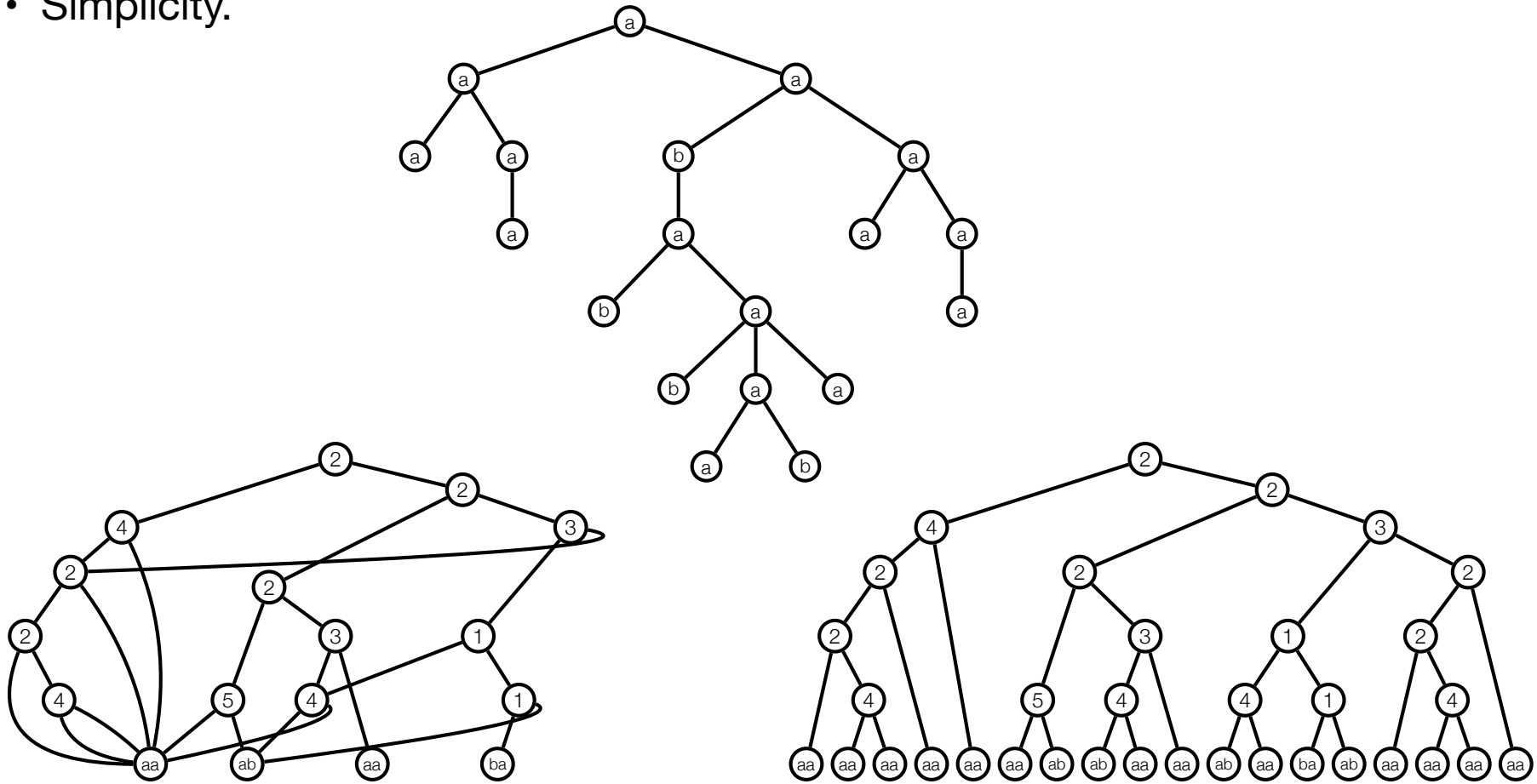

a
a
a
a
a
b
a
a
a
a
b
a
a
b
a
a
a
b


2
2
4
3
2
2
2
3
1
4
5
4
1
aa
ab
aa
ba


2
2
4
3
2
2
2
3
1
2
4
4
5
4
4
1
4
aa
aa
aa
aa
aa
aa
ab
ab
aa
aa
ab
aa
ba
ab
aa
aa
aa
aa

Top Tree Compression (2013)

- Why top tree compression?
  - Optimal tree compression.
  - Direct support for compressed computation.
  - Simplicity.



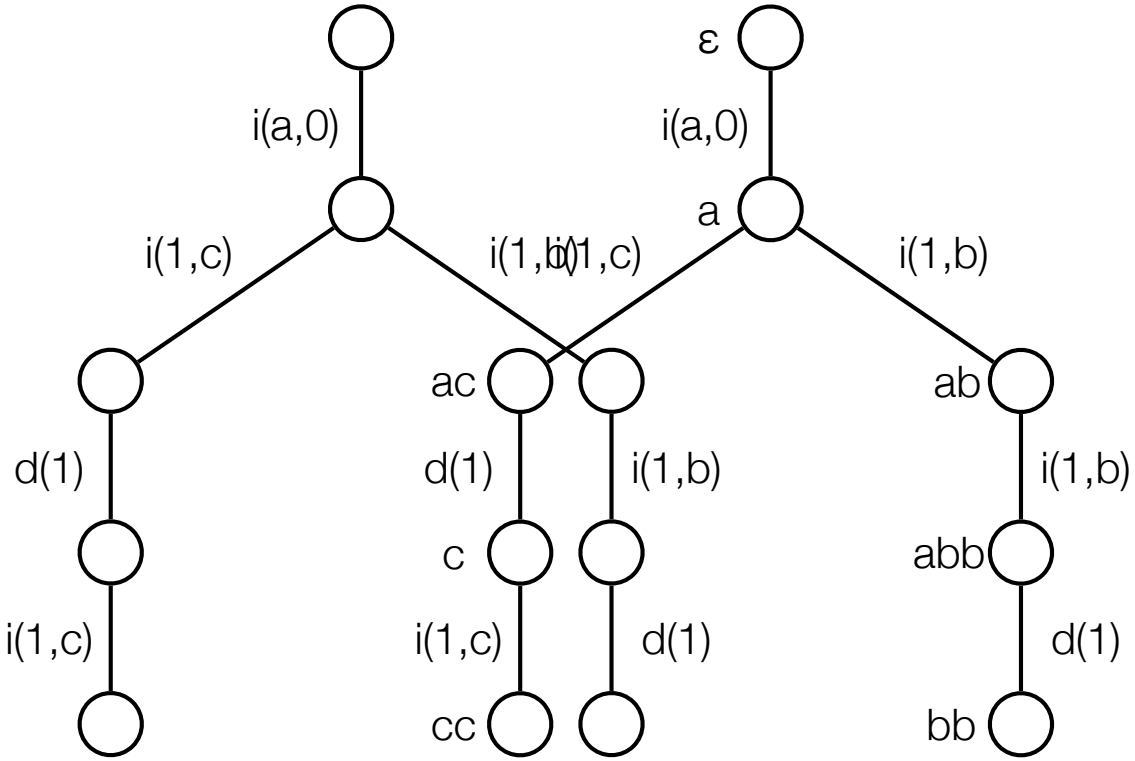
Highly-Repetitive Collections

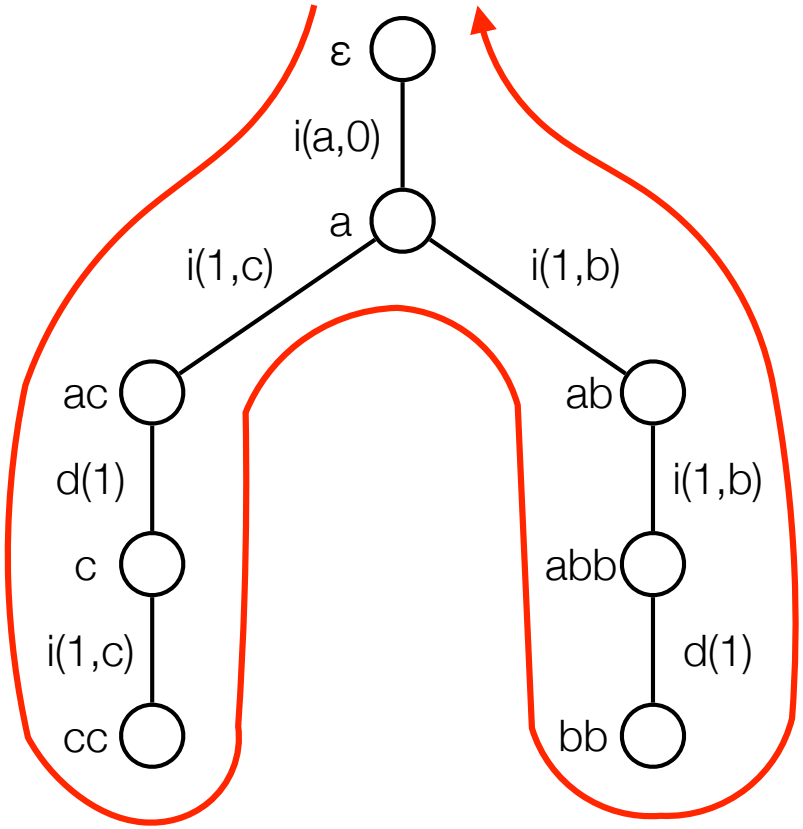
Highly-Repetitive Collections

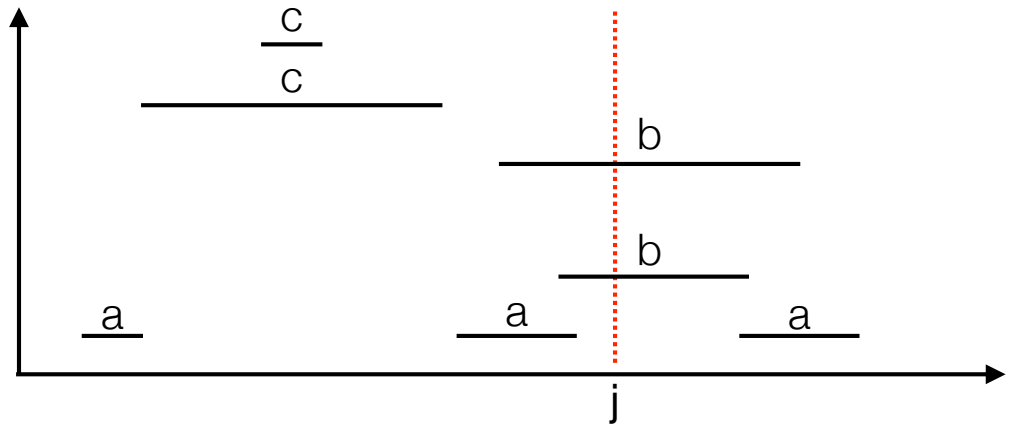
- Some data sets are collections of very similar of data items: genome data bases, versioned data.
- Goal: Compress highly-repetitive collections and support efficient compressed computation.

Persistent Strings (2020)

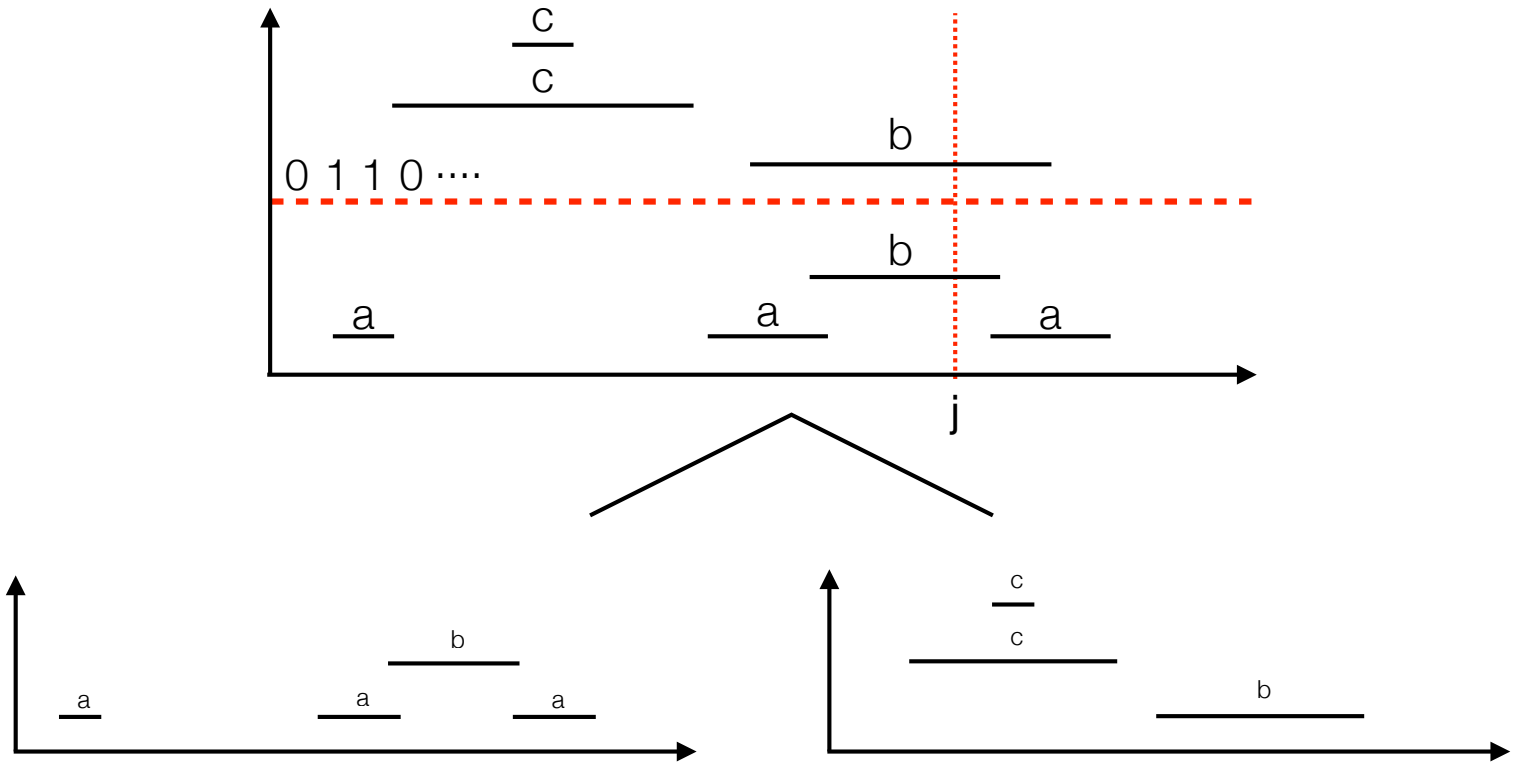
- Collections of strings represented by a version tree.
- Each node represents string. Obtained by applying edit operation on parent.
- What is the ith character of string at node j?







what is ith smallest segment at time j?

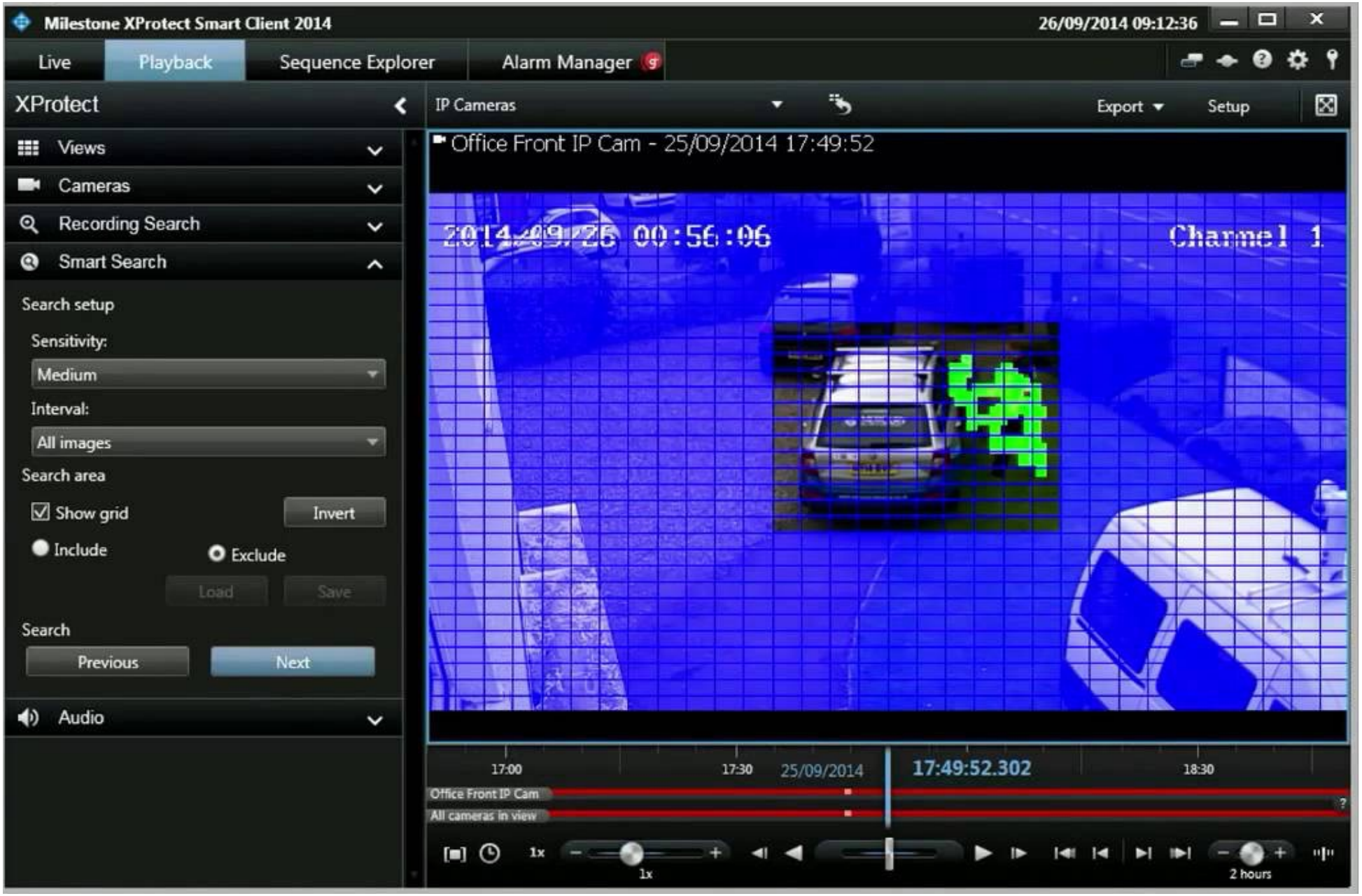


~~~log n steps~~

$$\sim \frac{\log n}{\log \log n} \text{ steps}$$

optimal
~~~

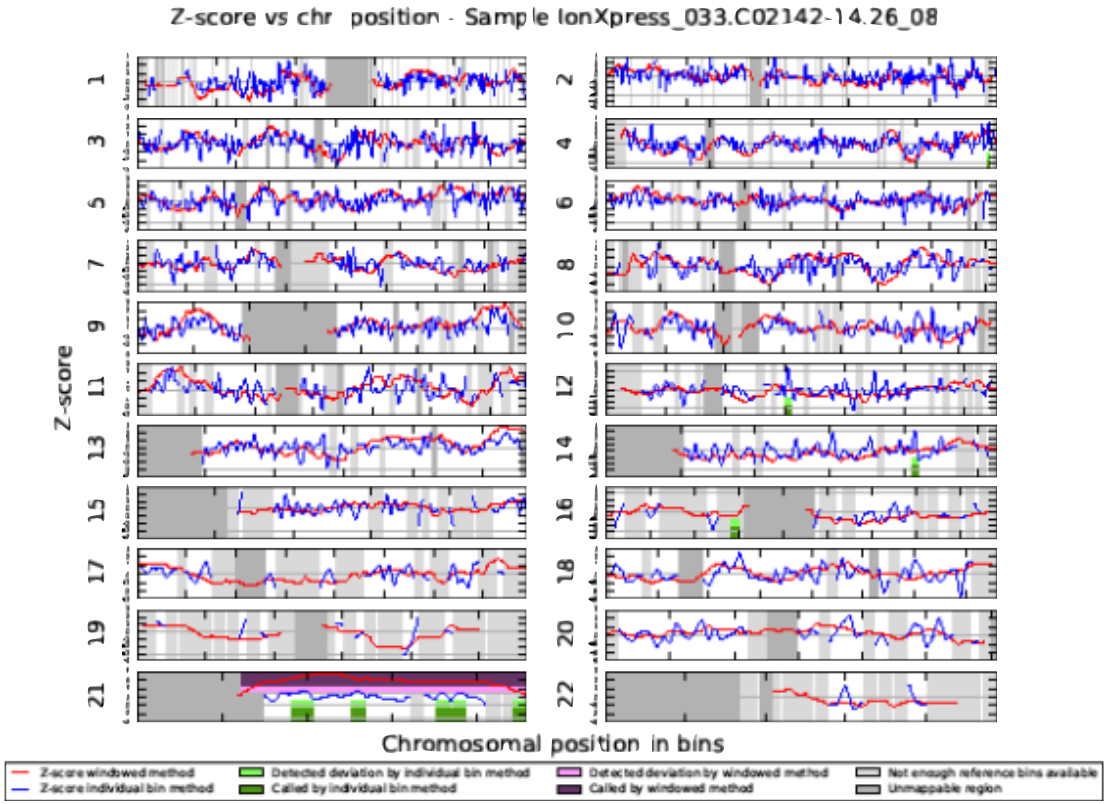
Applications

Video Surveillance

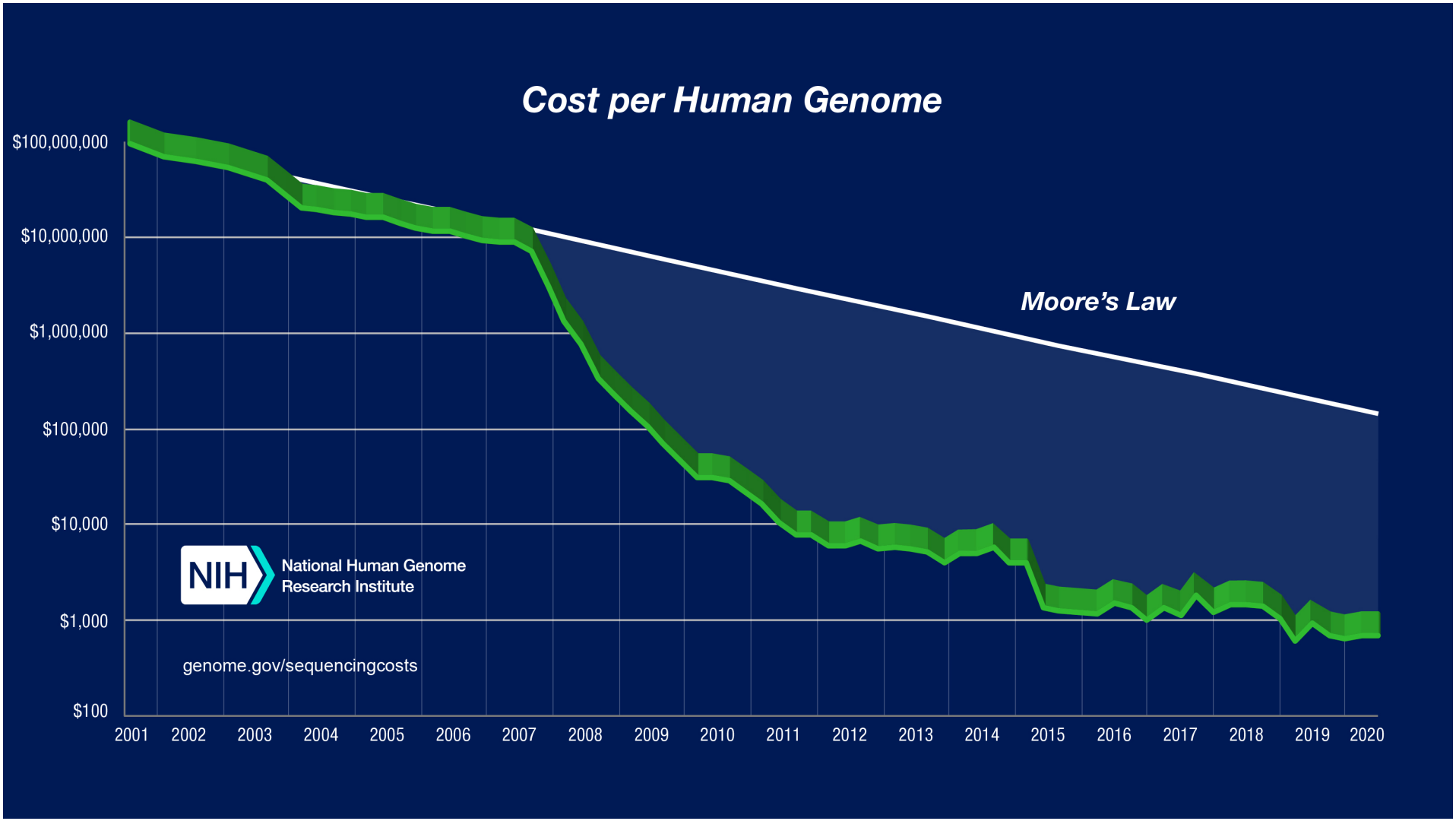
Video Surveillance



Next-Gen Sequencing



Next-Gen Sequencing



Thank You!