

Suffix Vector: Space- and Time-Efficient Alternative To Suffix Trees

Krisztián Monostori

Arkady Zaslavsky

Heinz Schmidt

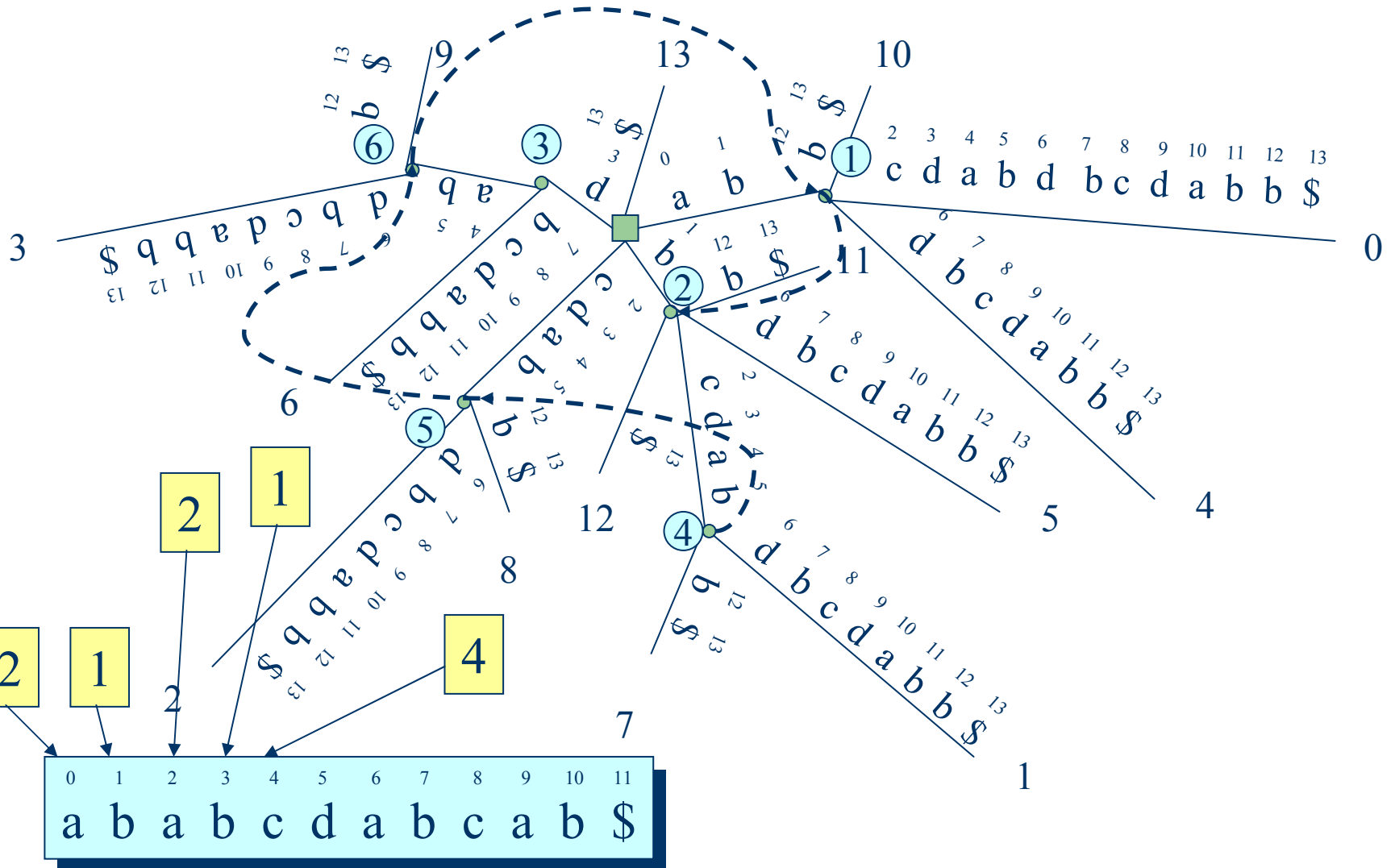
School of Computer Science and Software Engineering,
Monash University

Overview

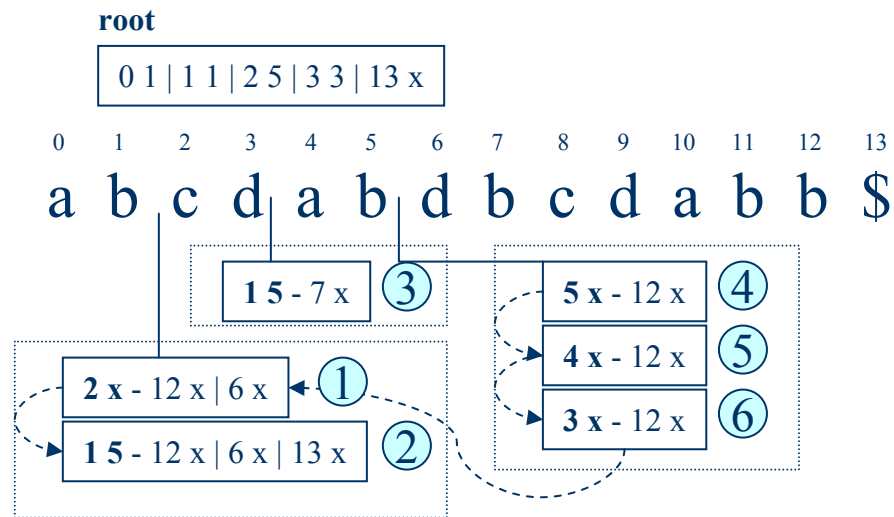
- Suffix tree representation
 - Matching statistics algorithm
- Suffix vector representation at a high level
- Physical suffix vector representation
- Performance results
- Algorithms on suffix vectors
- Conclusion and future work

Suffix Tree

0 1 2 3 4 5 6 7 8 9 10 11 12 13
a b c d a b d b c d a b b \$



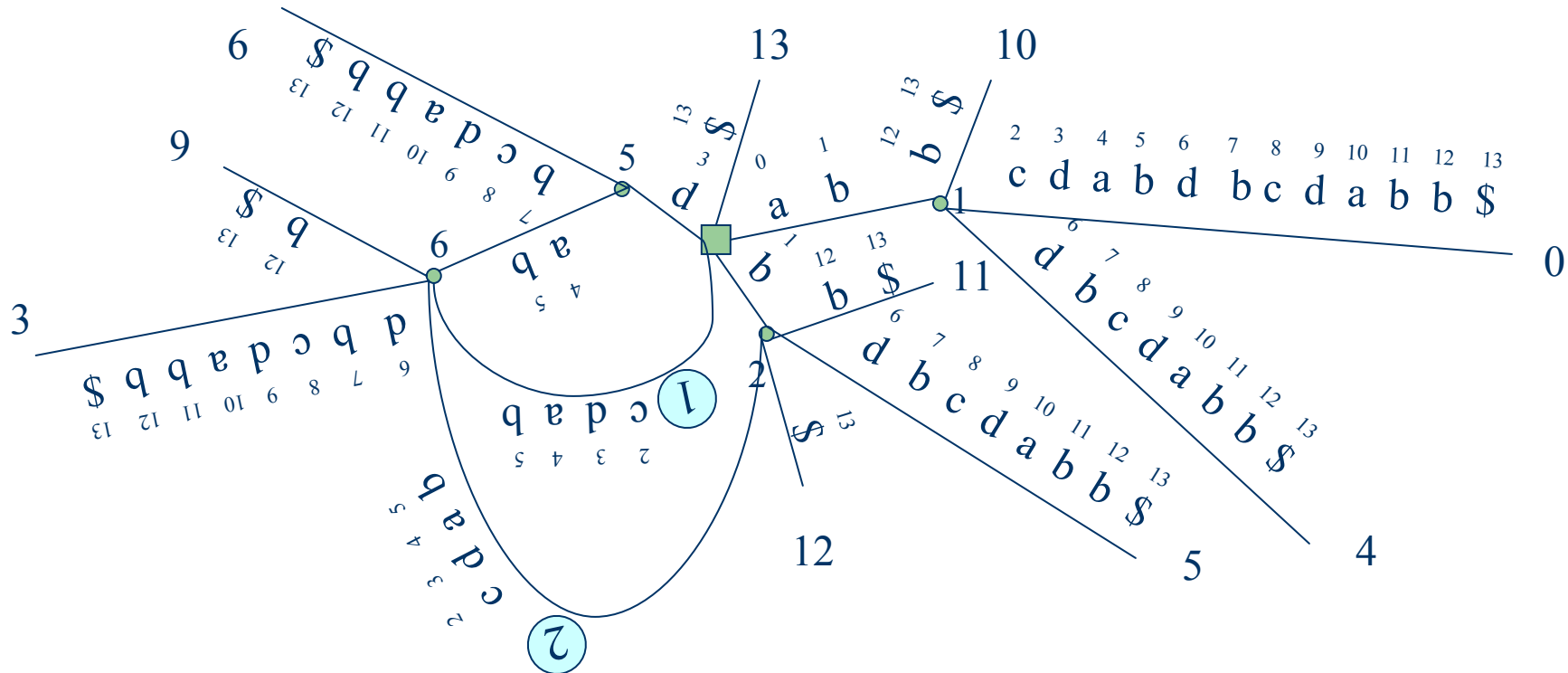
Suffix Vector at a High Level



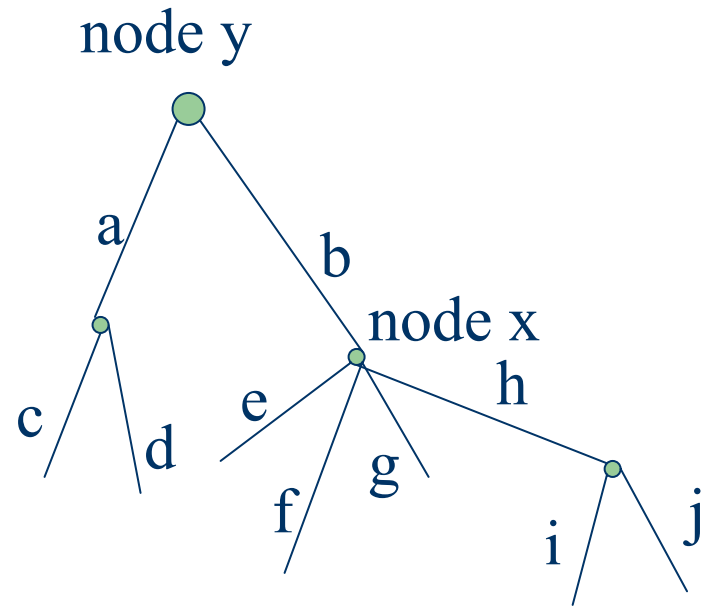
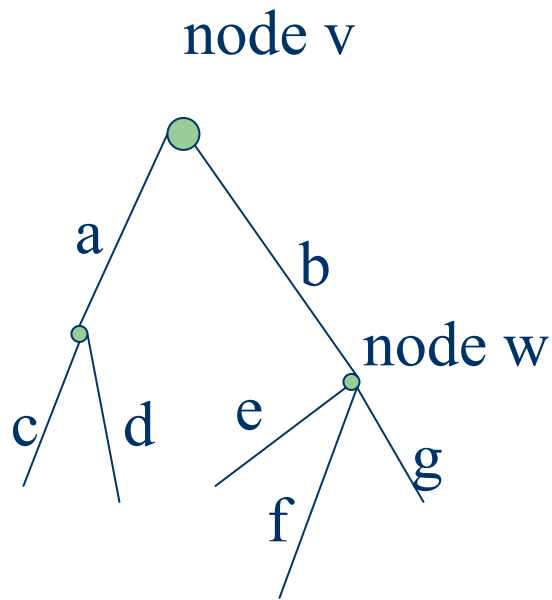
0	1	2	3	4	5
b	c	d	a	b	b

Directed Acyclic Graph

0	1	2	3	4	5	6	7	8	9	10	11	12	13
a	b	c	d	a	b	d	b	c	d	a	b	b	\$

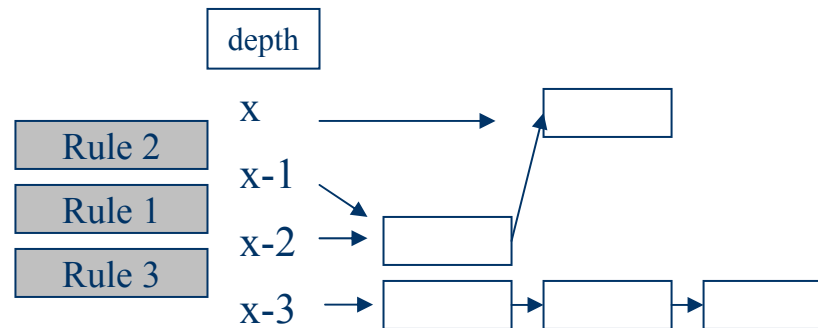


Redundancy in Suffix Trees

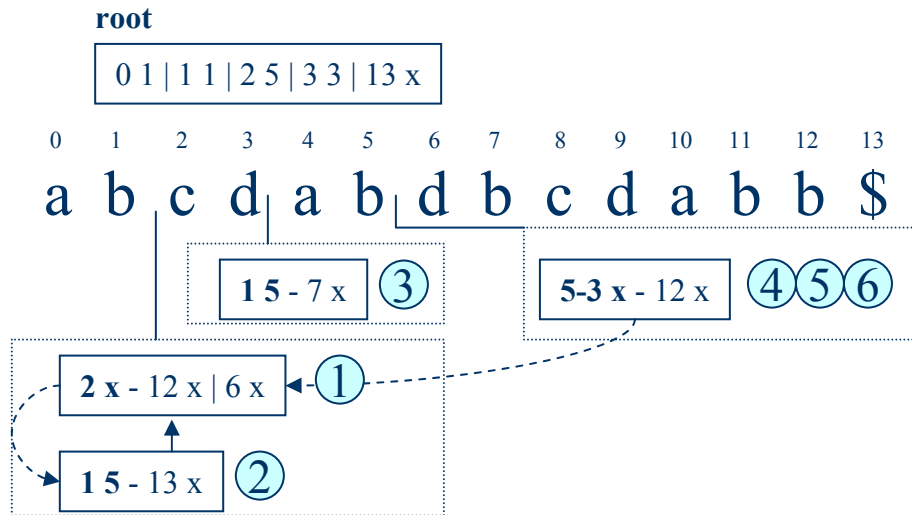


Edge Sharing Rules

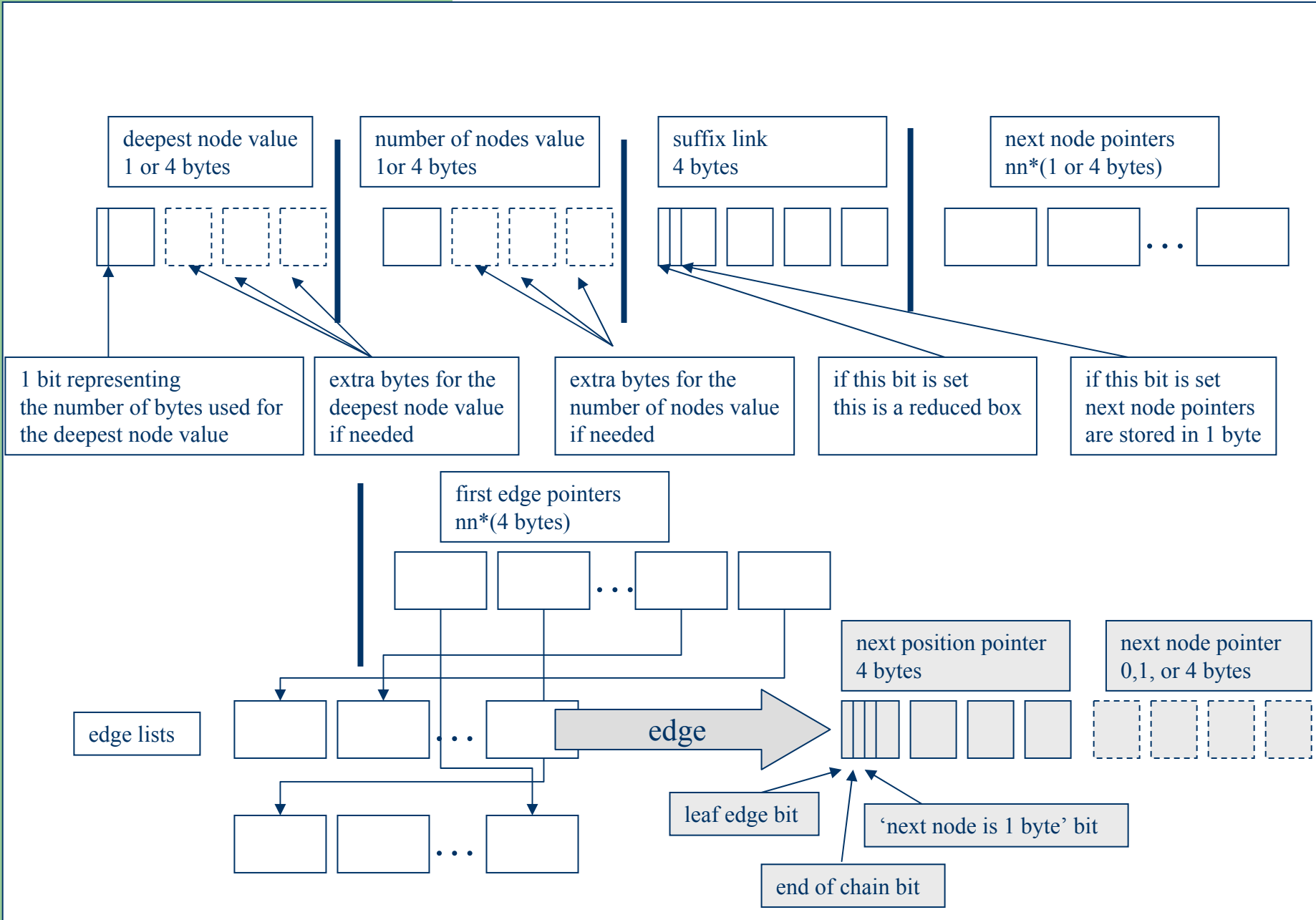
- **Rule 1:** same edges
- **Rule 2:** same edges plus extra edges
- **Rule 3:** not the same edges



Eliminating Redundancies



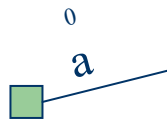
Physical Representation of a Box



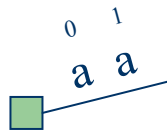
Performance Results

File name:	File size	Vector size	Bytes/symbol Suffix Vector	Bytes/symbol Kurtz
book2	610856	5561505	8.61	9.67
progl	71646	603927	8.06	10.22
K02402	38095	484229	12.56	12.59

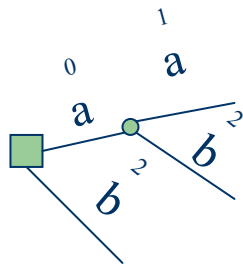
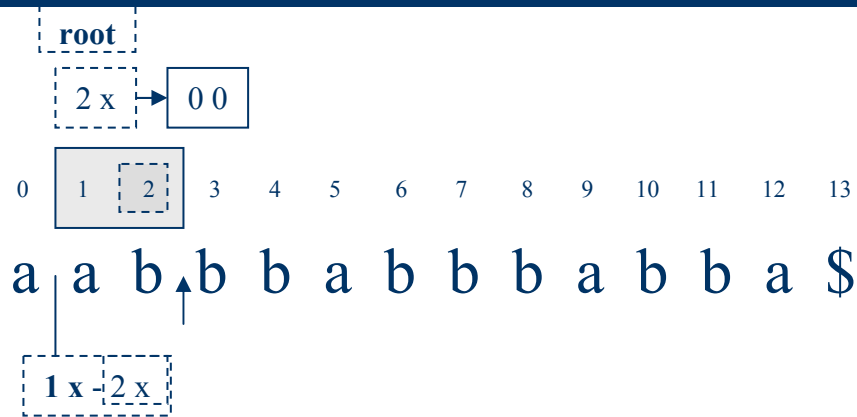
Suffix Vector Construction



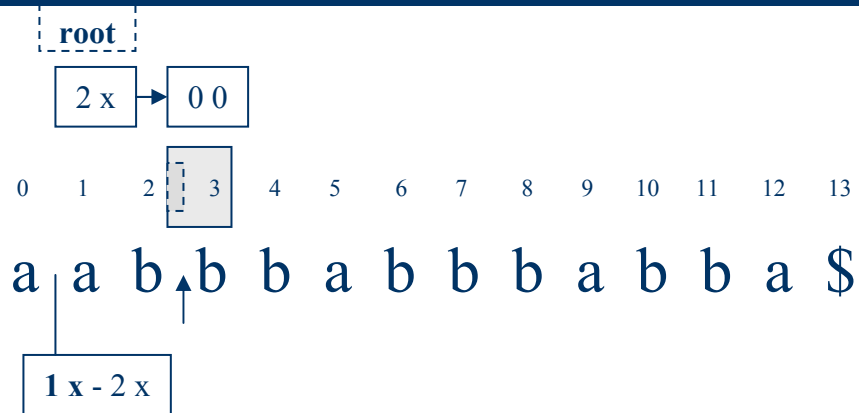
Suffix Vector Construction



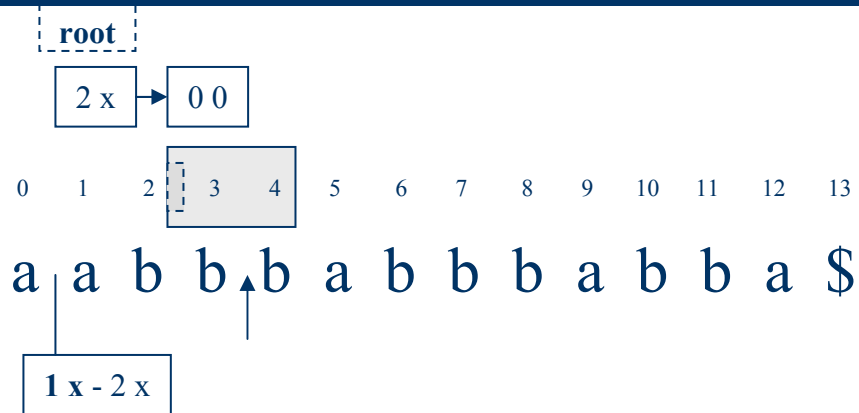
Suffix Vector Construction



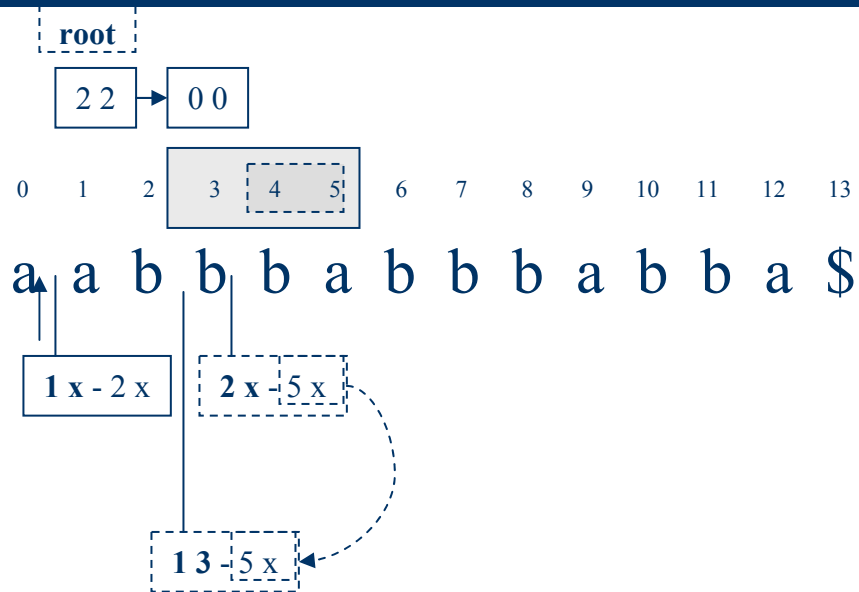
Suffix Vector Construction



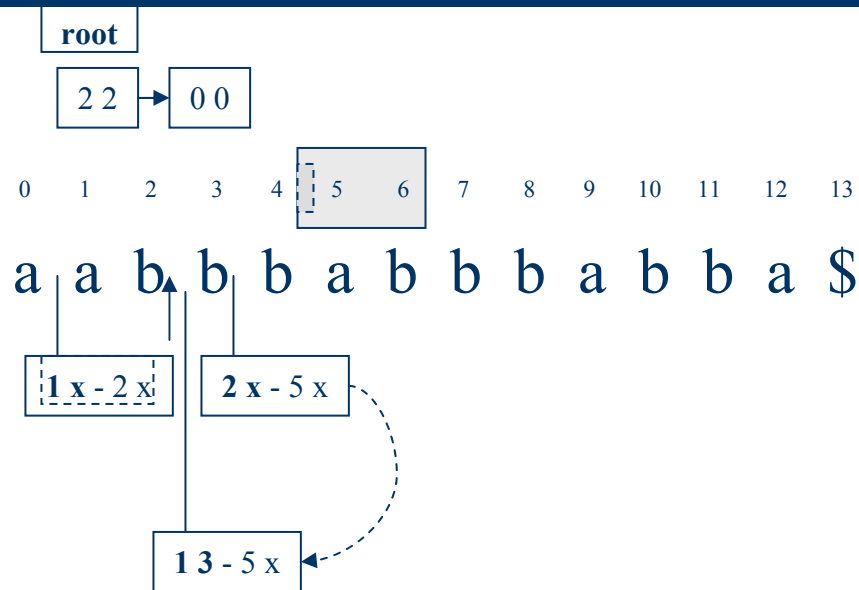
Suffix Vector Construction



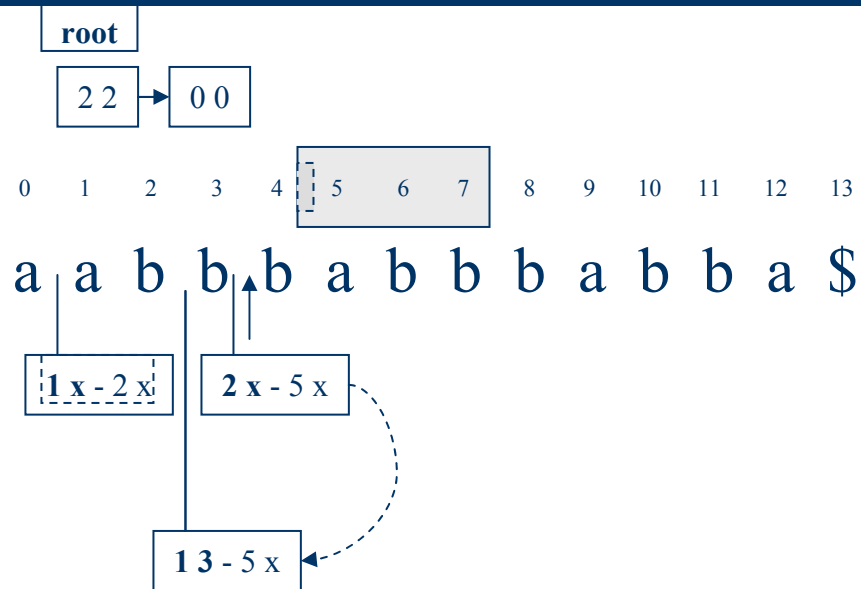
Suffix Vector Construction



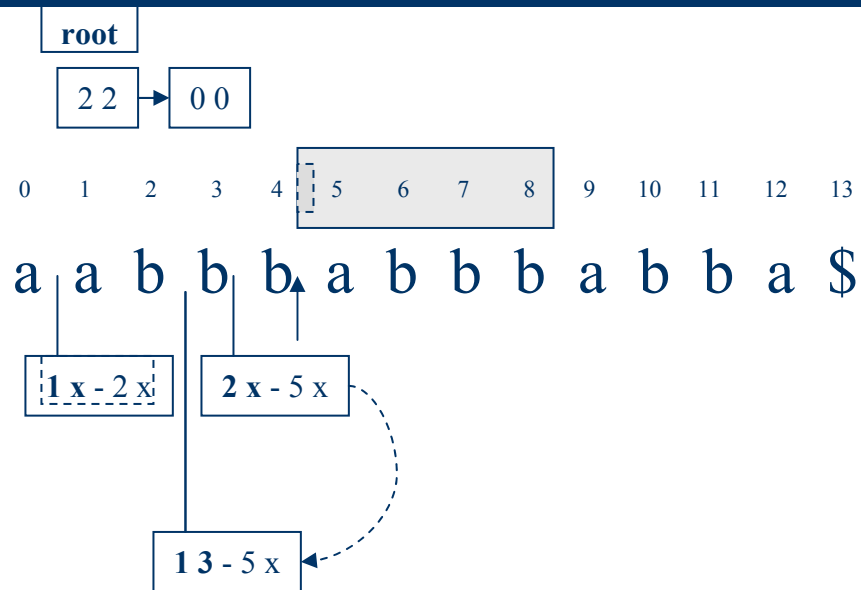
Suffix Vector Construction



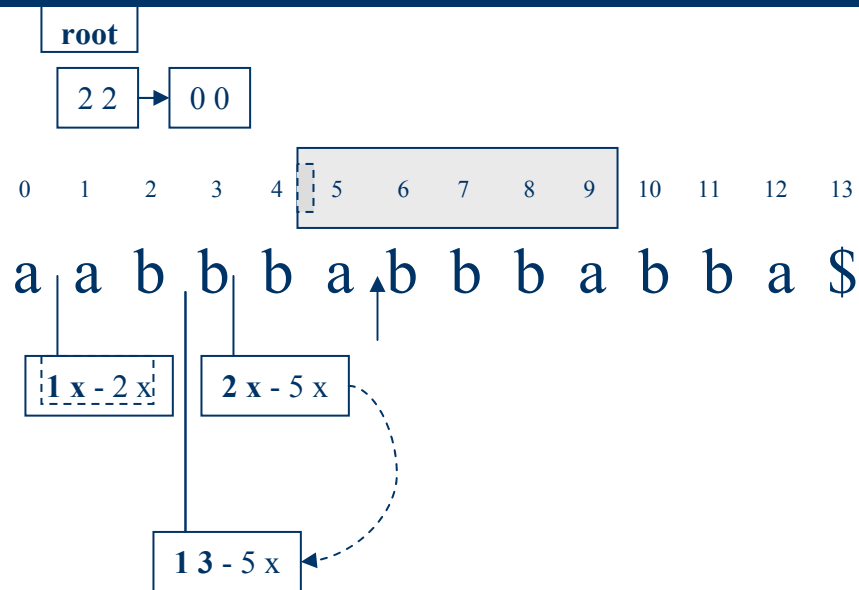
Suffix Vector Construction



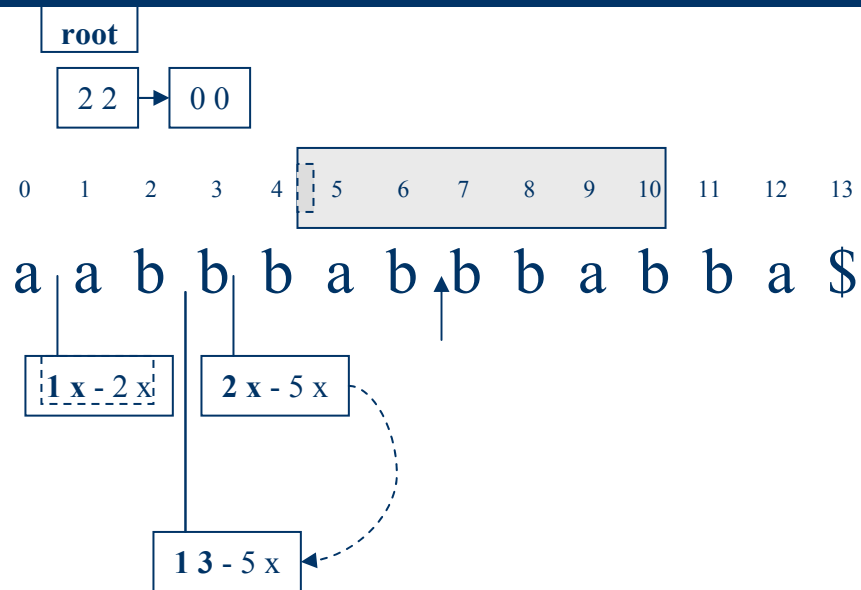
Suffix Vector Construction



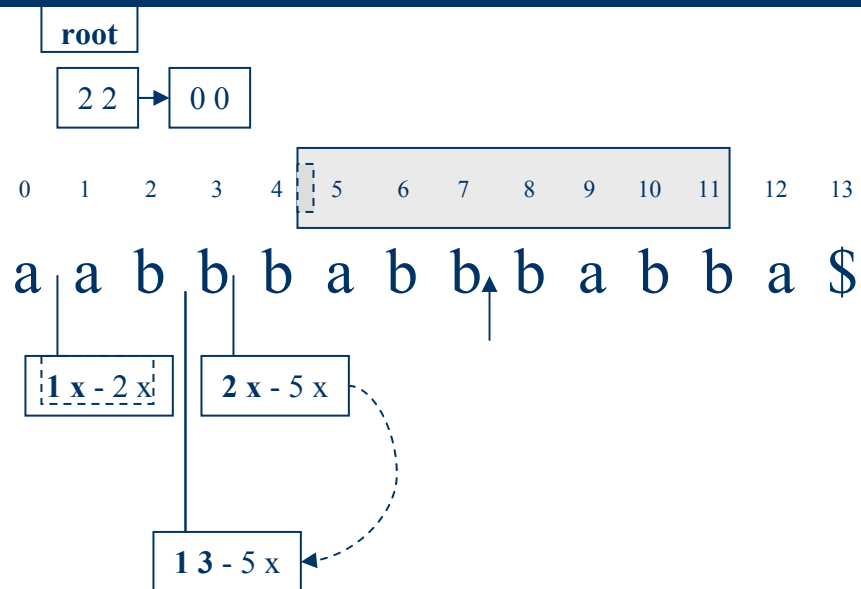
Suffix Vector Construction



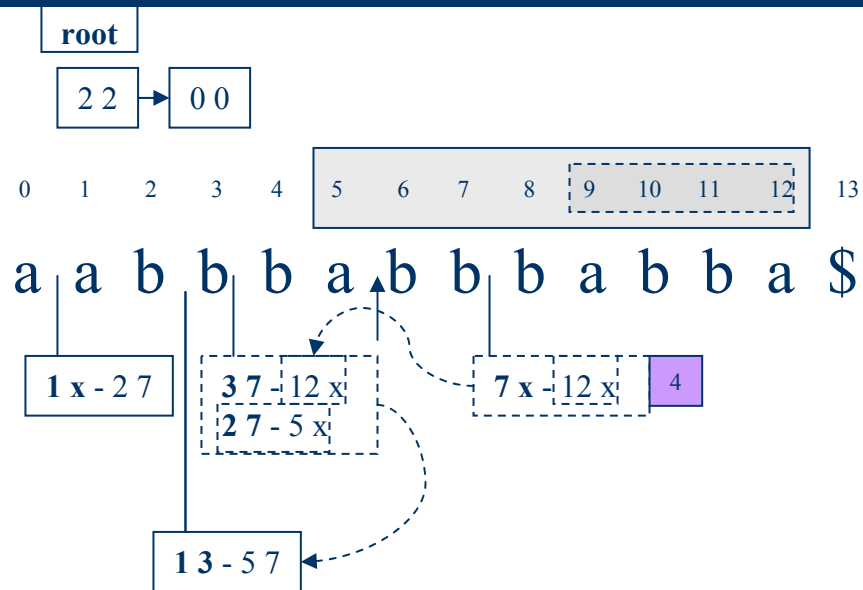
Suffix Vector Construction



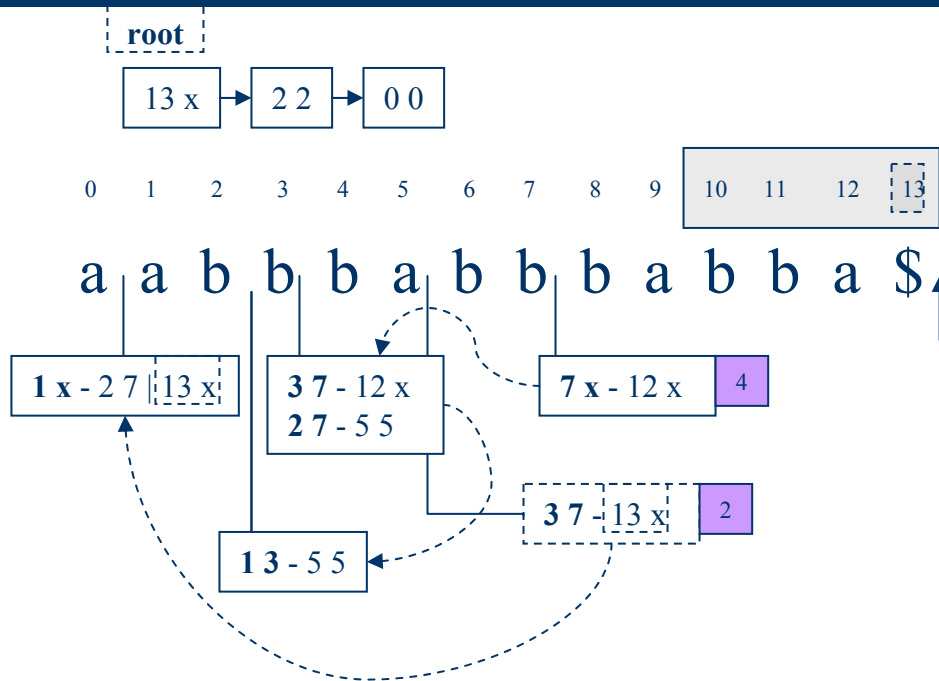
Suffix Vector Construction



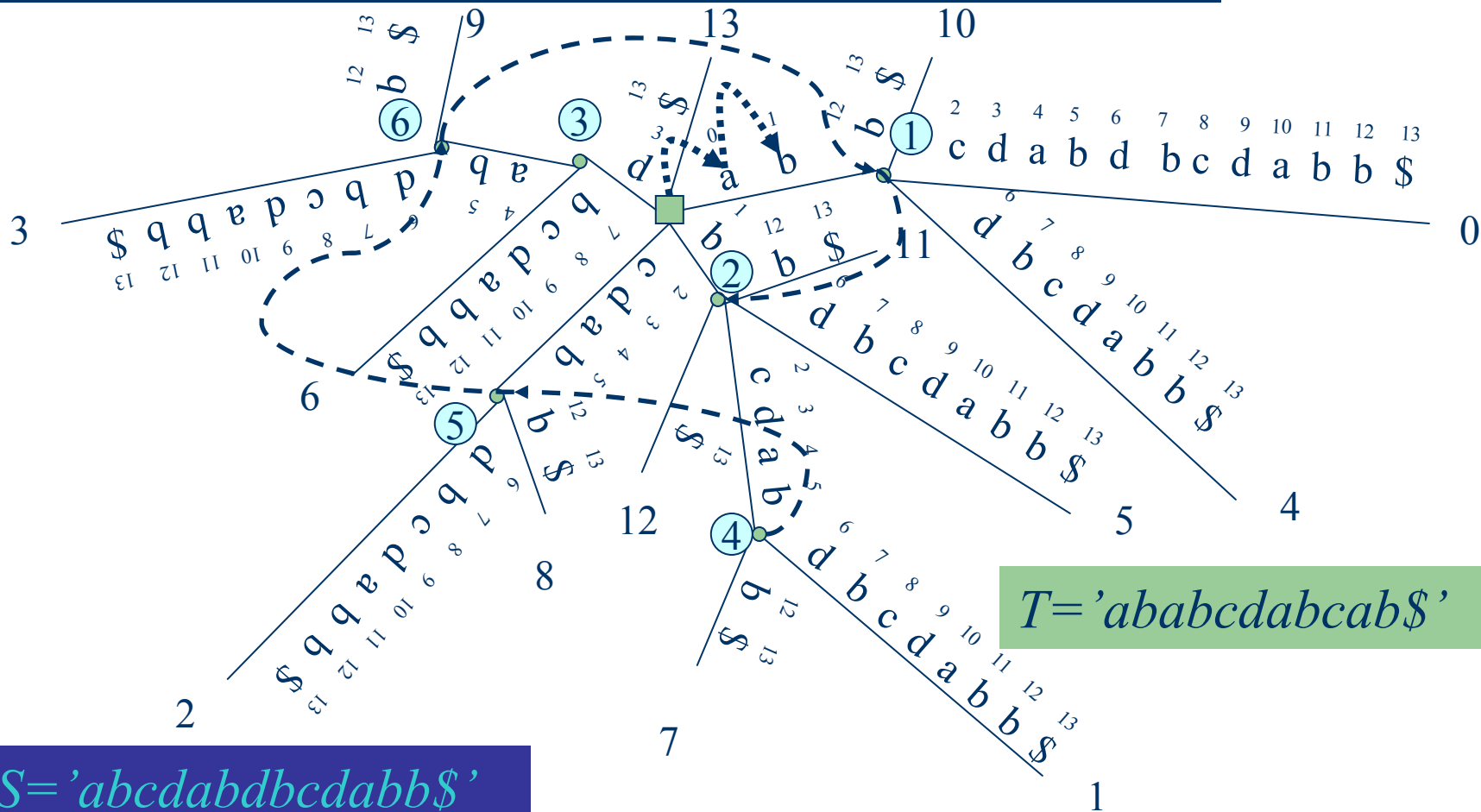
Suffix Vector Construction



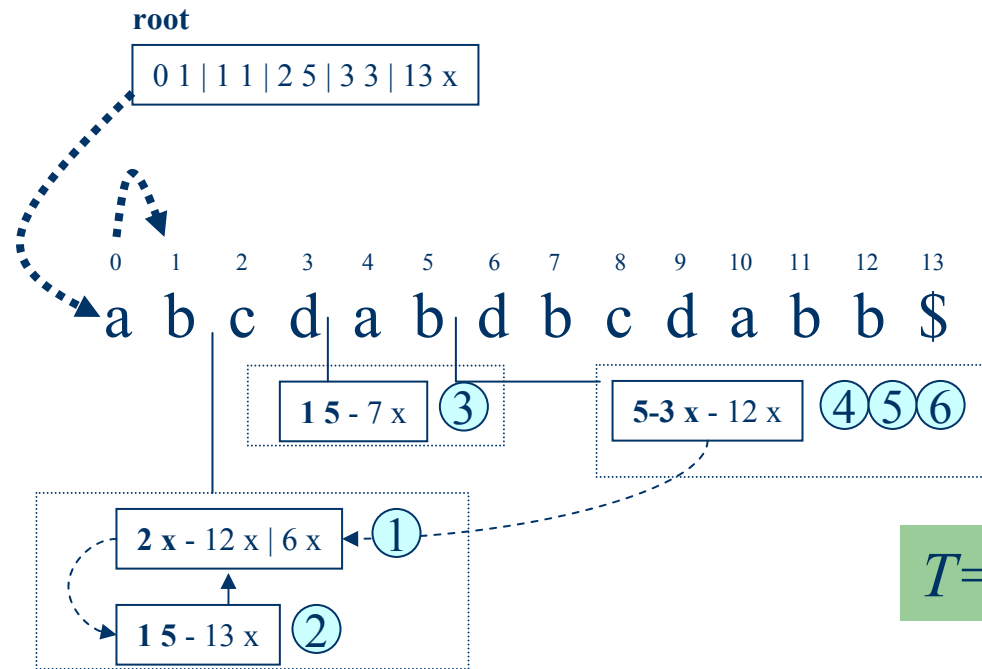
Suffix Vector Construction



Matching Statistics Algorithm on a Suffix Vector



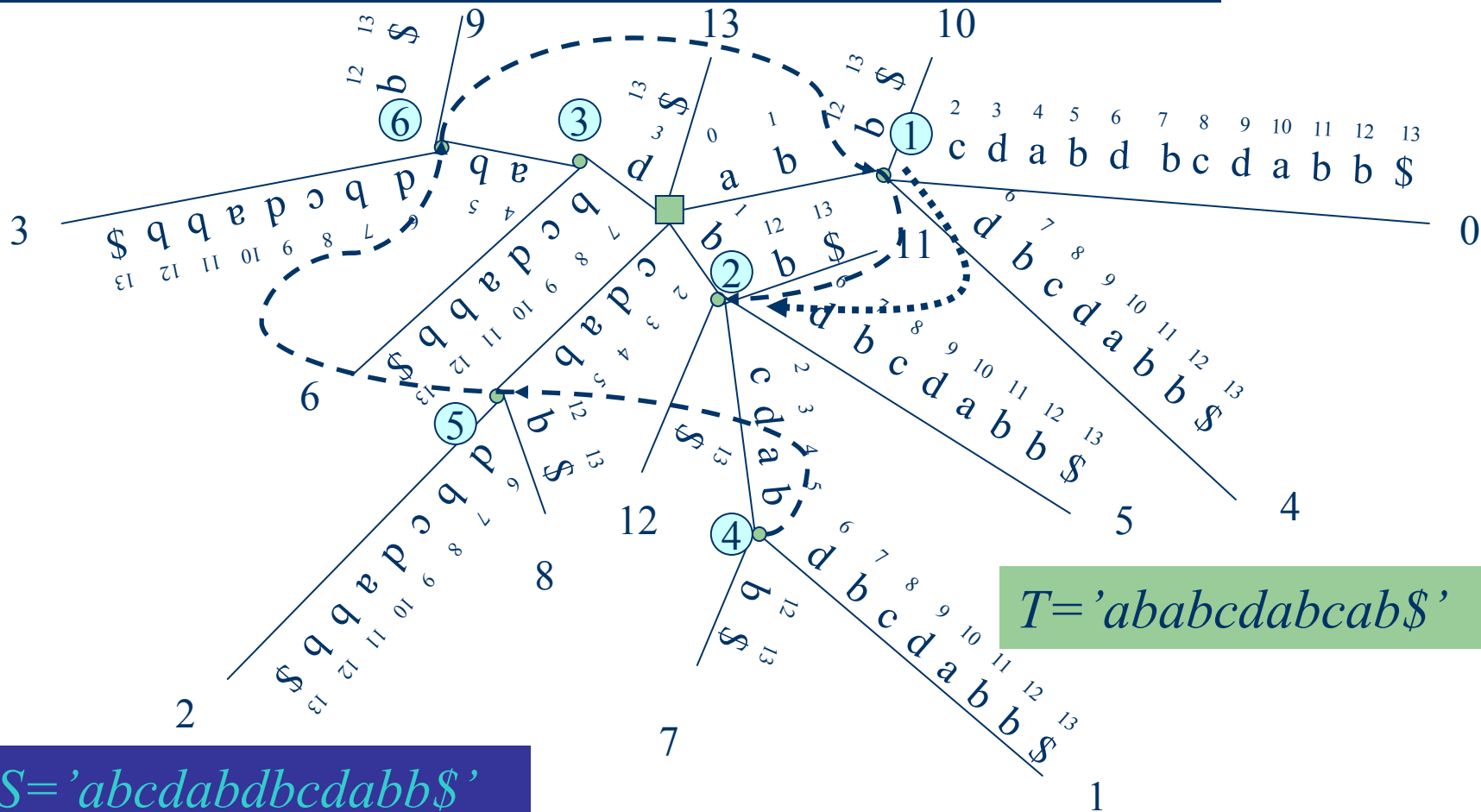
Matching Statistics Algorithm on a Suffix Vector



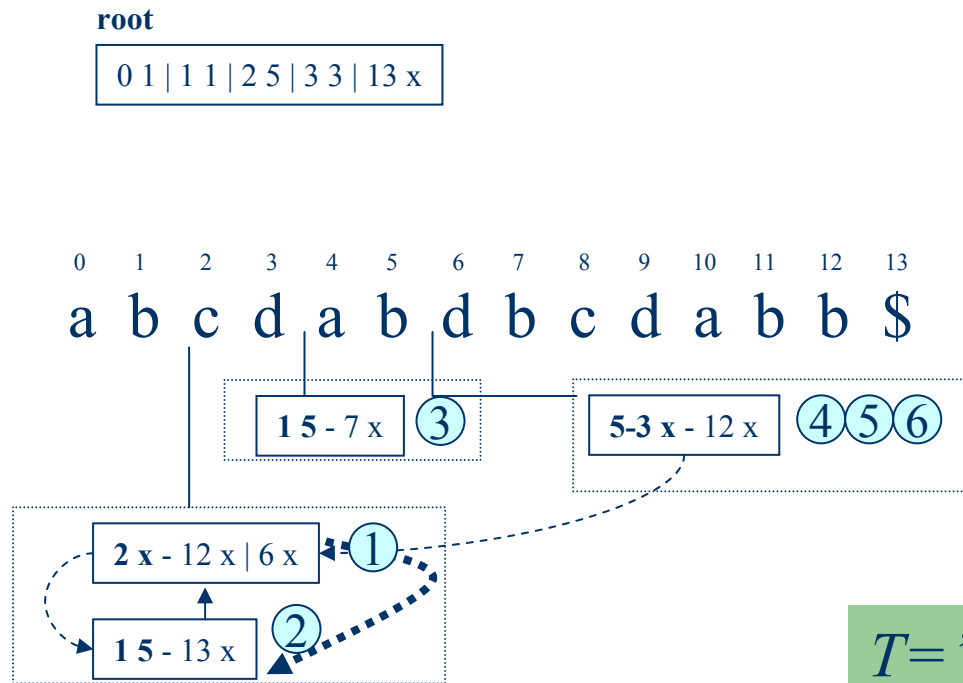
$T = 'ababcdabcb\$'$

$S = 'abcdabdbcdabb\$'$

Matching Statistics Algorithm on a Suffix Vector



Matching Statistics Algorithm on a Suffix Vector



$T = 'ababcdabcbab\$'$

$S = 'abcdabdbcdabb\$'$

Conclusion and Future Work

- Most space-efficient representation
- Direct construction: extra space
- Matching statistics algorithm runs faster
- Future work
 - Other algorithms
 - Other physical representations