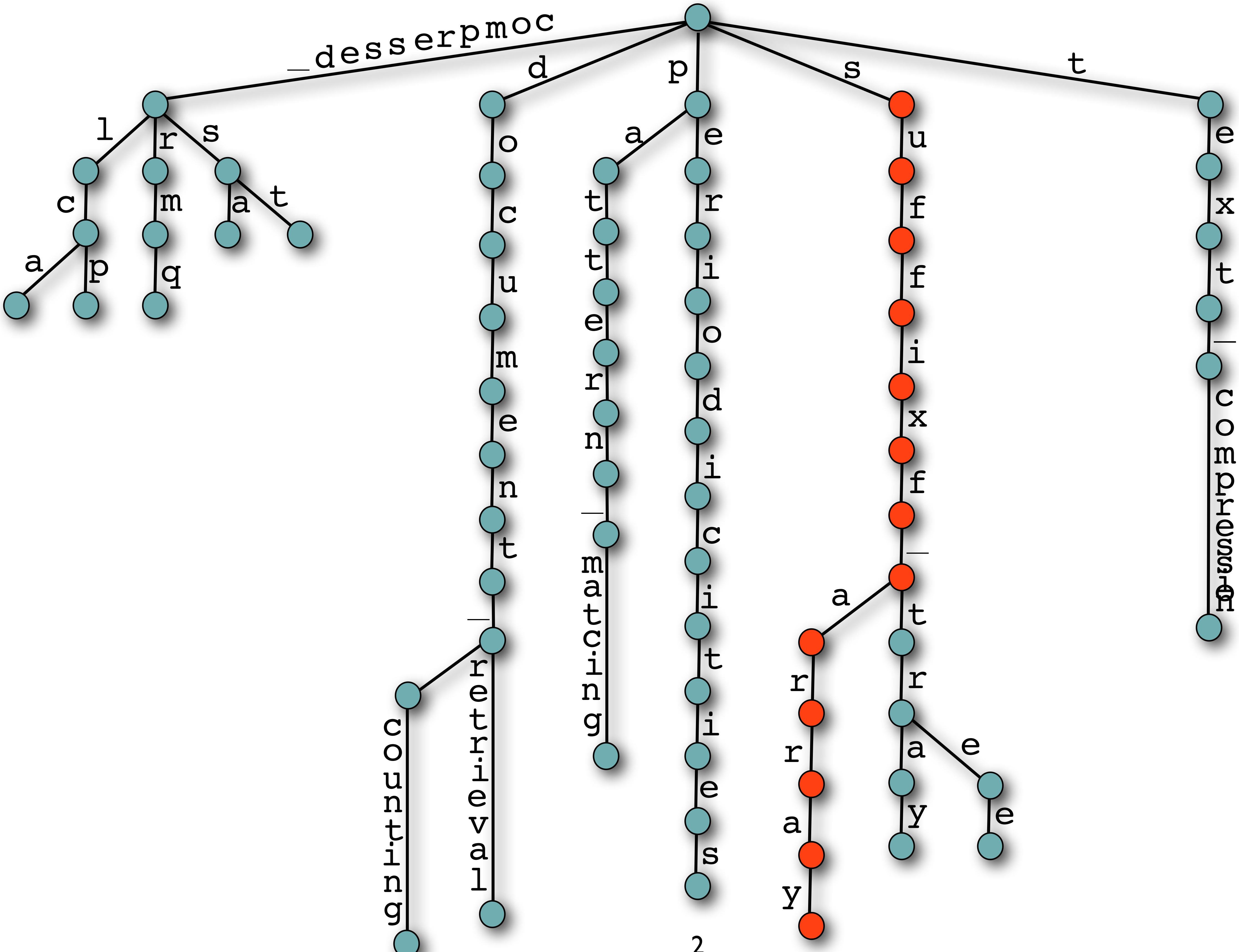
Lecture 4: LCP Arrays Range Minimum Queries

Johannes Fischer

Topics

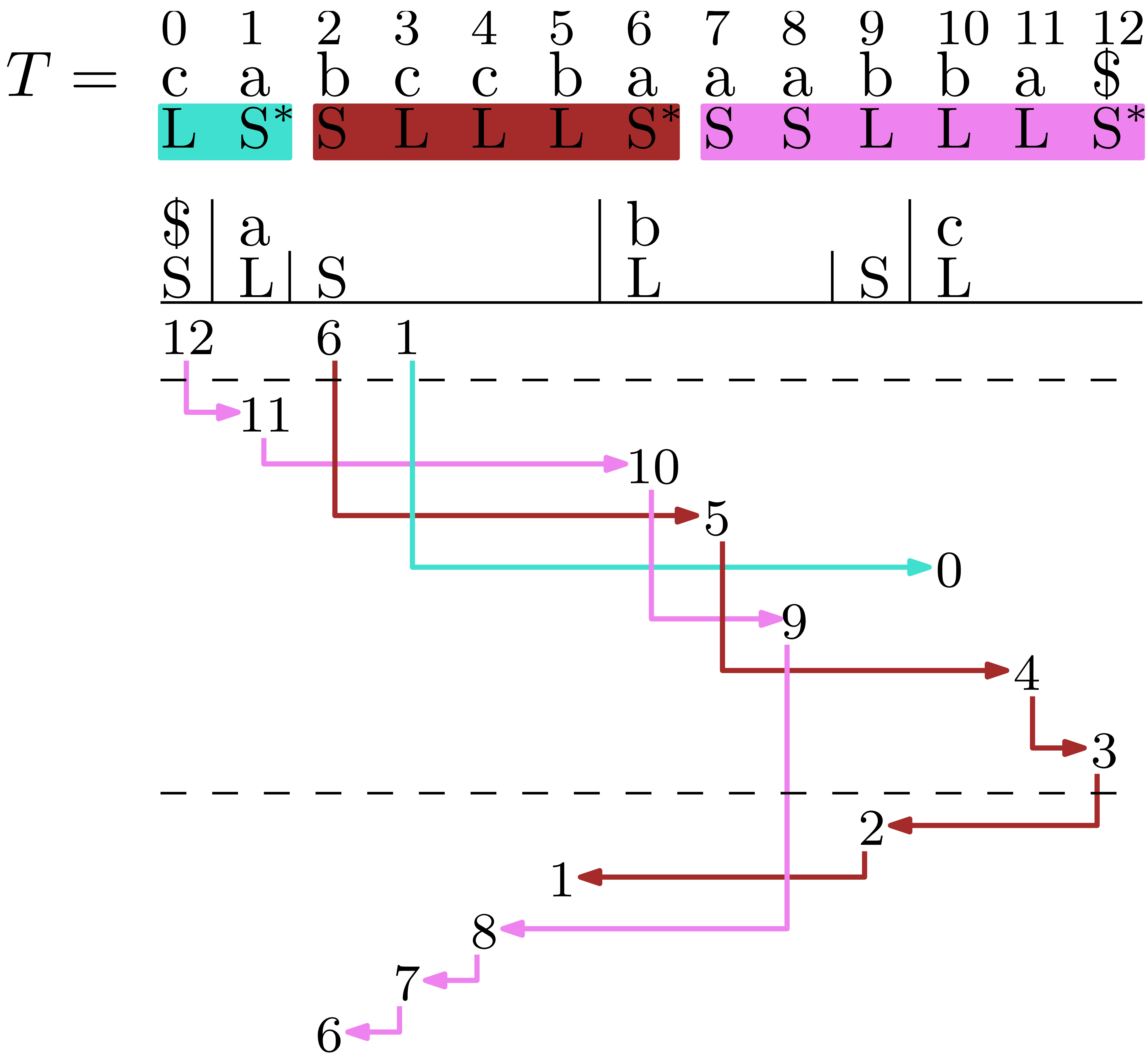


_desserpmoc
d
p
s
t
counting
retrieval
matching
compression

Algorithm sais

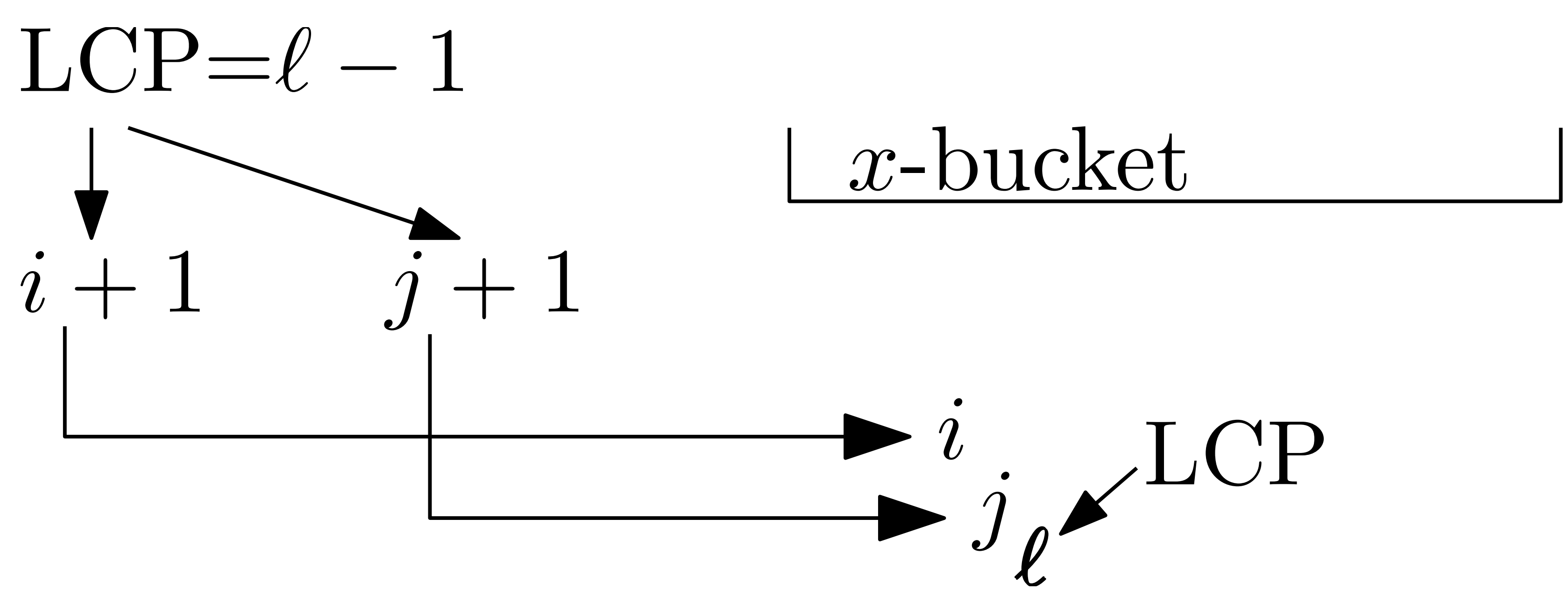
- Definition: suffix $T[i,n]$ called
  - **S-type** iff $T[i..n] <_{lex} T[i+1..n]$ (T[n,n]='$' always S)
  - **L-type** otherwise

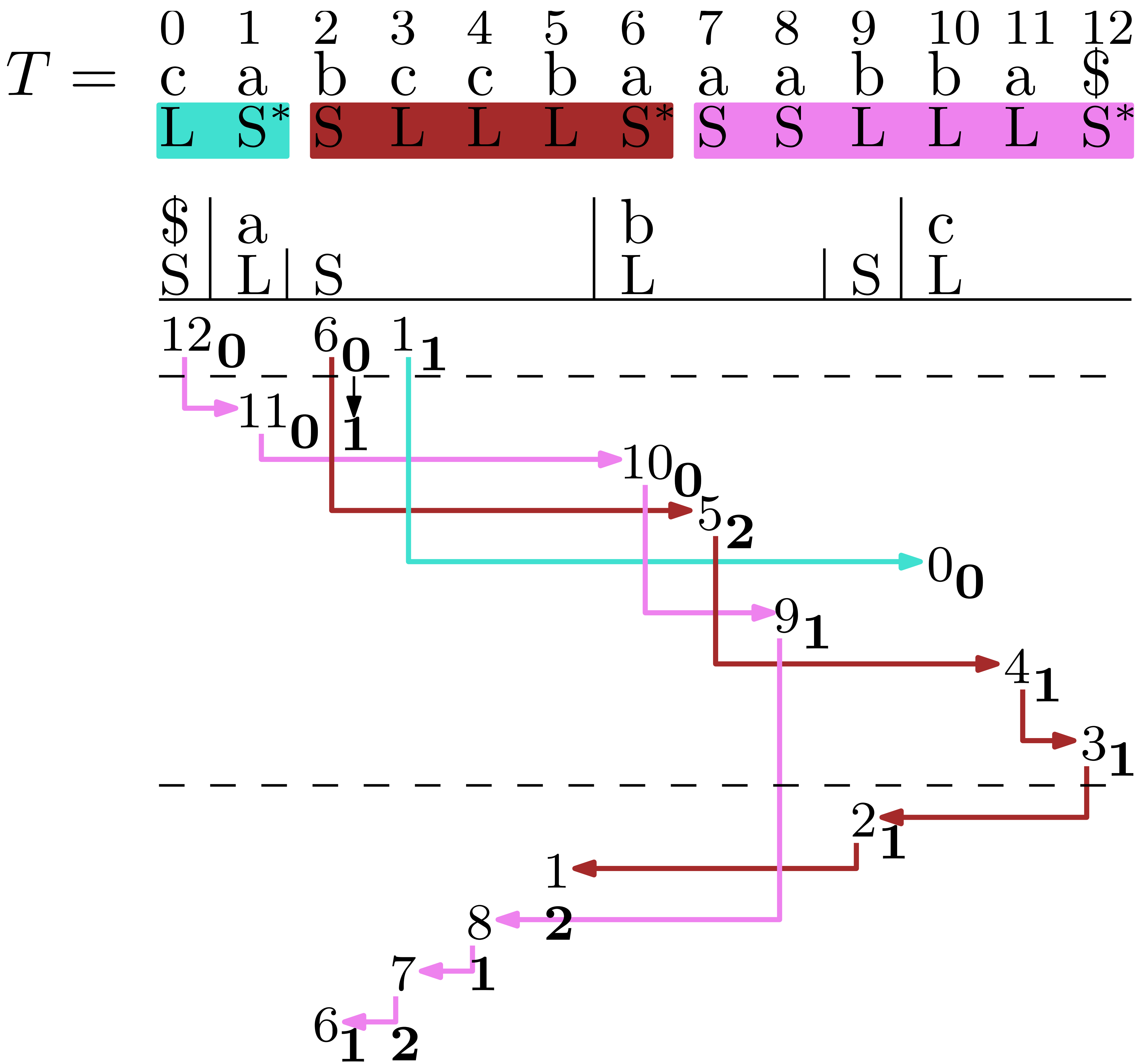
1. Choose sample: leftmost S (call them S*), $|S^*|<1/2n$
2. Sort S^*-suffixes by **recursion**
   - on new text formed by sorted S^*-substrings
3. Scan A from left to right (say we're at pos. i):
   - if $T[A[i]-1]$ is **L**, write $A[i]-1$ to 1st pos. in bucket
4. like (3), but sorting **S**-suffixes in a right-to-left scan

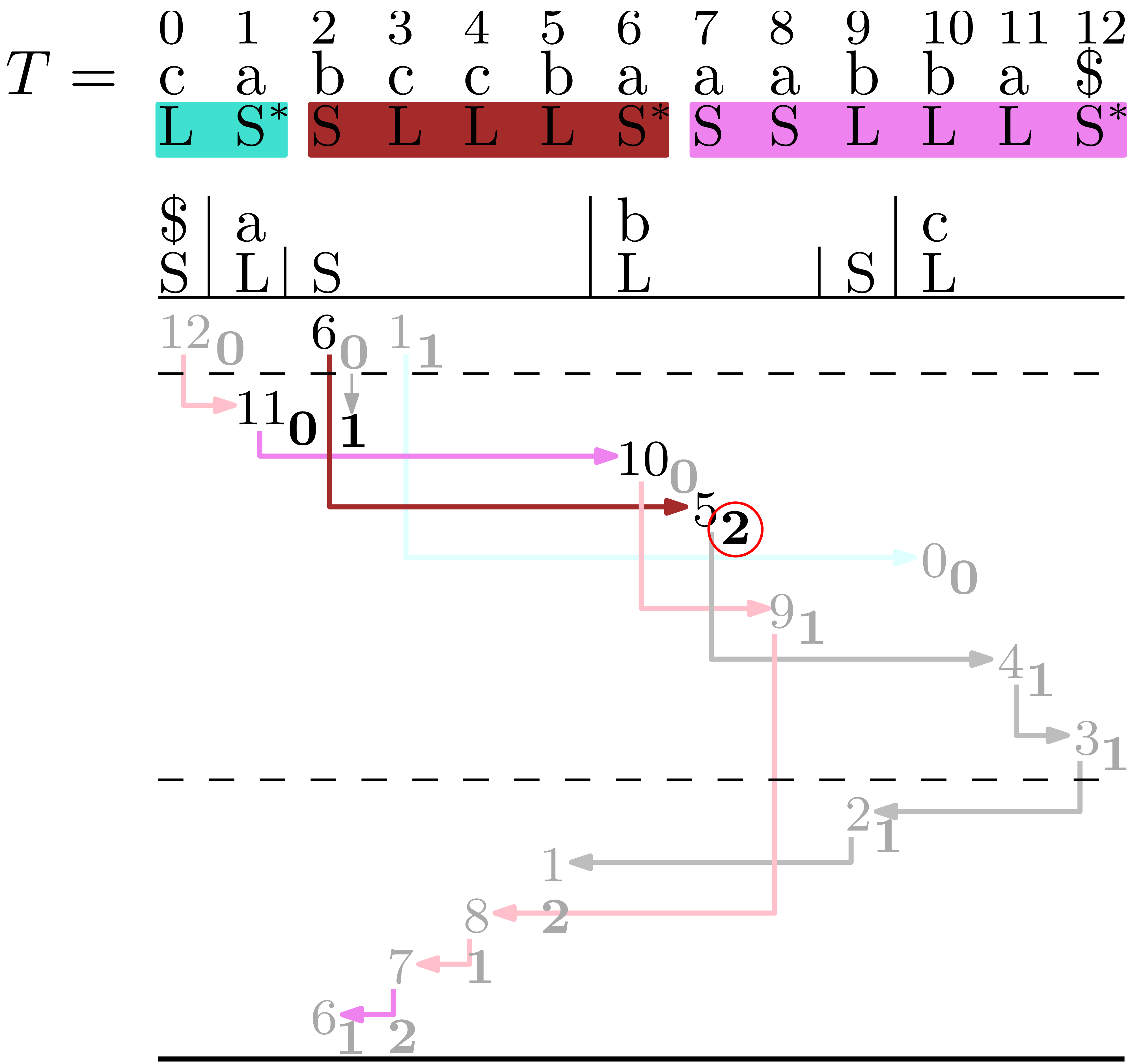

T =
0 1 2 3 4 5 6 7 8 9 10 11 12
c a b c c b a a a b b a $
L S* S L L L S* S S L L L S*
$ a b c
S L S L S L
12 6 1
11
10
5
0
9
4
3
2
1
8
7
6

Inducing LCPs

- **Idea**
  - placing *T*[*i*..*n*] and *T*[*j*..*n*] at adjacent positions...
  - ... allows us to induce LCP ...
  - ... from LCP of *T*[*i*+1..*n*] and *T*[*j*+1..*n*]...
  - ... which is already known!

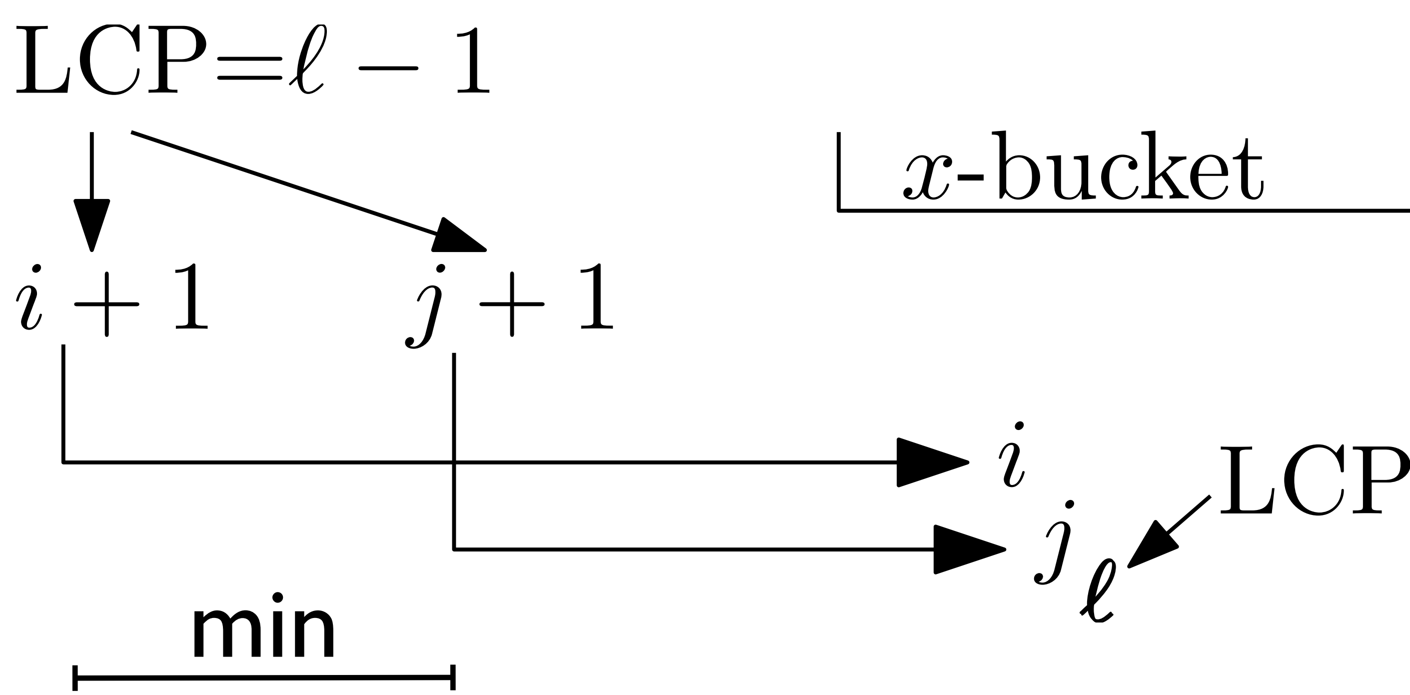



T =
0 1 2 3 4 5 6 7 8 9 10 11 12
c a b c c b a a a b b a $
L S* S L L L S* S S L L L S*
$ a b c
S L S L S L
12 0 6 0 1 1
11 0 1
10 0
5 2
0 0
9 1
4 1
3 1
2 1
1
8 2
7 1
6 1 2


T =
0 1 2 3 4 5 6 7 8 9 10 11 12
c a b c c b a a a b b a $
L S* S L L L S* S S L L L S*
$ a b c
S L S L S L

One Detail

- What is LCP of suffixes *i*+1 and *j*+1?



- In general:
  $LCP(T[x..n],T[y..n])=LCP[RMQ_{LCP}(SA^{-1}[x], SA^{-1}[y])]$