An Efficient Algorithm for Identifying the Most Contributory Substring

Ben Stephenson
Department of Computer Science
University of Western Ontario

Outline

- Problem Definition
- Related Problems
- Applications
- Algorithm
- Summary
- Questions

Problem Definition

- Given a set of strings, the most contributory sub-string, *w*, is a string of characters such that the length of *w*, denoted by |*w*|, times the number of times it occurs is maximal
- Equivalently, the most contributory sub-string, *w*, is the string that reduces the number of characters in the set by the greatest amount when all occurrences of *w* are removed

Related Problems

- Longest Common Substring Problem
  - Finds the longest string common to all strings in the set
  - Does not consider repeated occurrences
- All Maximal Repeats Problem
  - Finds the maximal substring that occurs twice within a string

Applications

- Analyzing Profile Data
  - Determine what sequence of events occurs frequently within the profile data
- Data Compression
  - What sub-string should be represented by the shortest code word?
- Bioinformatics
  - Find common sub-strings in DNA sequences

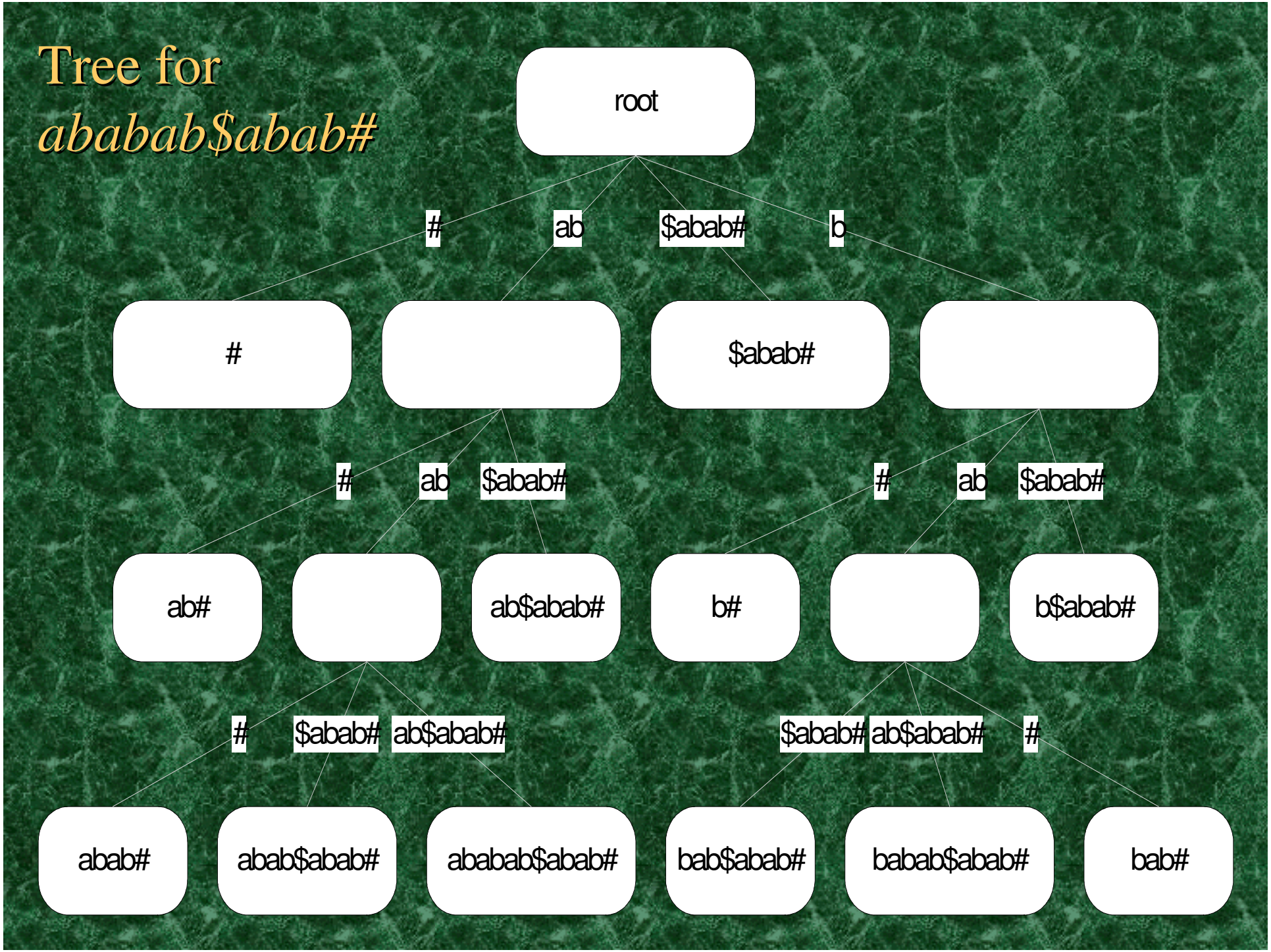
Algorithm

- Suffix Trees
  - Data structure used to efficiently implement many string algorithms
  - Can be constructed for a set of strings by concatenating the strings and placing unique sentinel characters between each string
  - Can be constructed in linear time with respect to the length of the string

Suffix Tree Properties

- Tree contains n leaf nodes
  - Total nodes in the tree is $O(n)$
- Each branch in the tree is labelled with a string
  - Collecting the strings on the branches as one traverses the tree from the root to a leaf forms a suffix of the string represented by the tree
- Each non-leaf node has at least 2 children

Tree for *ababab$abab#*

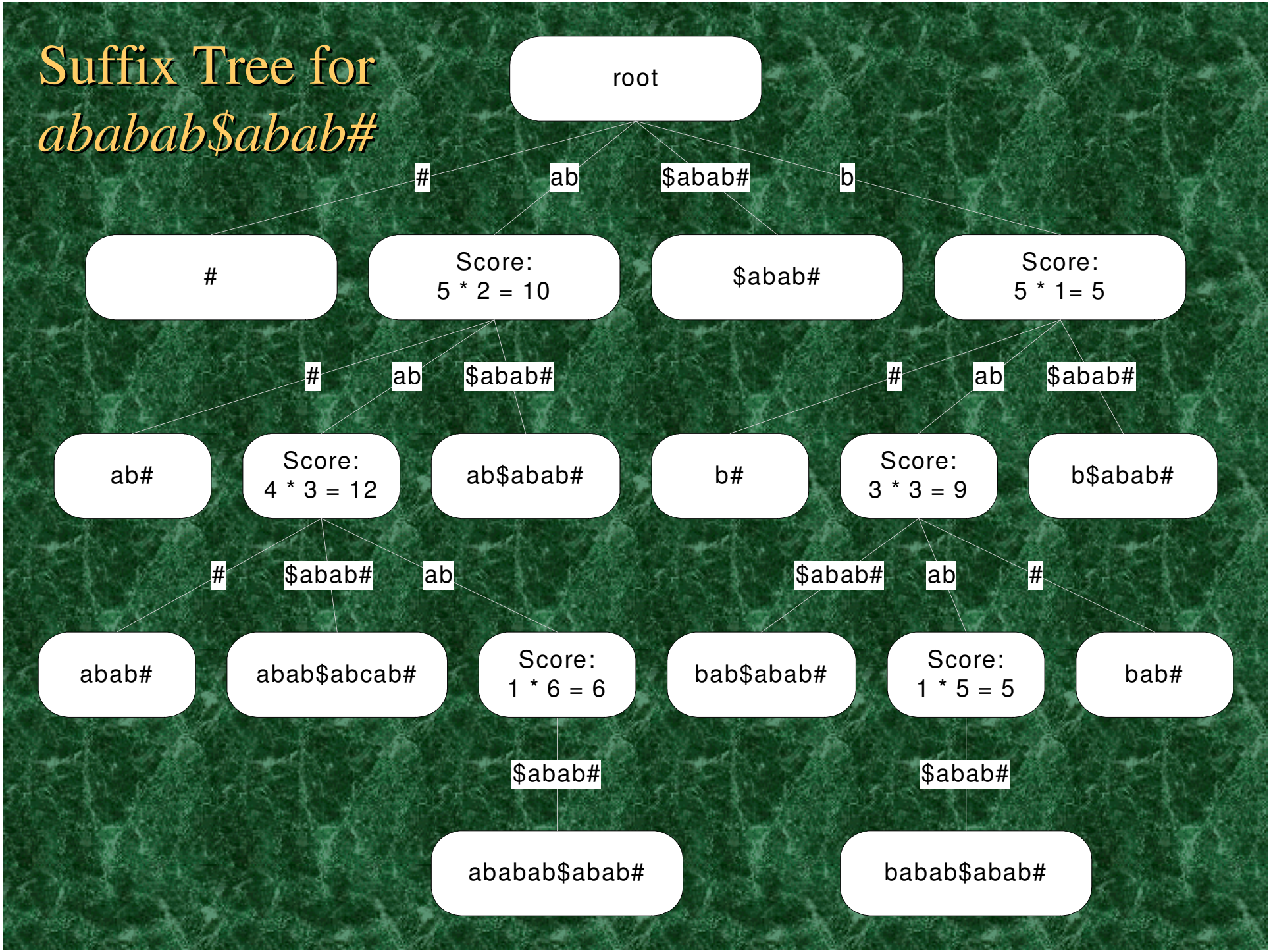


Finding the Most Contributory Substring

- Split Leaf Nodes
  - Any leaf node that is reached by a branch that starts with a sentinel character is left untouched
  - Other leaf nodes are split into two nodes
    - New leaf node is reached by a branch starting with a sentinel character
    - New interior node has only one child
  - This can be done in $O(n)$ time

Finding the Most Contributory Substring

- Scoring Traversal
  - Determine the string depth of each node
    - The total number of characters along all branches traversed to reach the node
  - Determine the number of leaves below each node
  - The score of each node is computed as the product of its string depth the number of leaves below it
  - This can be done in O(n) time


Suffix Tree for
ababab$abab#
root
#
ab
$abab#
b
#
Score:
5 * 2 = 10
$abab#
Score:
5 * 1= 5
#
ab
$abab#
#
ab
$abab#
ab#
Score:
4 * 3 = 12
ab$abab#
b#
Score:
3 * 3 = 9
b$abab#
#
$abab#
ab
$abab#
ab
#
abab#
abab$abcab#
Score:
1 * 6 = 6
bab$abab#
Score:
1 * 5 = 5
bab#
$abab#
$abab#
ababab$abab#
babab$abab#

The Most Contributory Substring

- Highest Score: 12
- String Represented: *abab*

ababab$abab#

abab

abab

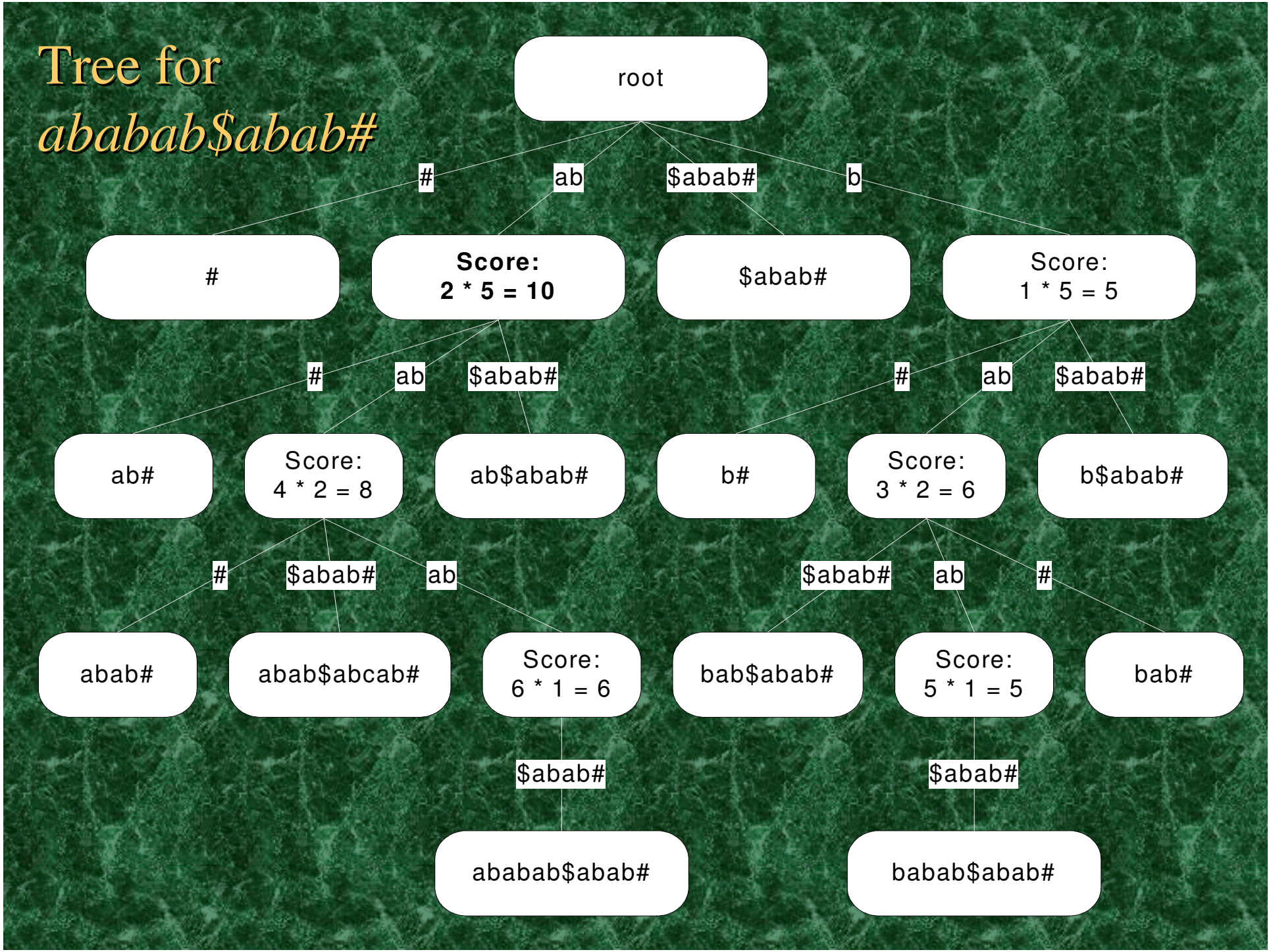
abab

- But two of the occurrences overlap…

Eliminating Overlapping Occurrences

- Two occurrences of a substring overlap if the difference in their start positions is less than their length
- Must determine the start positions of all substrings represented by each node and eliminate those that overlap
- Worst case: $O(n^2 \log(|\Sigma| + |L|))$
- Expected case: $O(n \log n \log (|\Sigma| + |L|))$

Tree for *abababab$abab#*



Most Contributory Substring

- Highest Score: 10
- String Represented: *ab*
- Occurrences of the substring do not overlap

Summary

- The Most Contributory Substring problem identifies the substring that represents the largest number of characters within a set of strings
- Applications:
  - Profile Data
  - Data Compression
  - Bioinformations

Summary

- Complexity
  - Overlapping substrings permitted
    - $O(n)$
    - Bounded by suffix tree construction time
  - Overlapping substrings not permitted
    - Expected: $O(n \log n \log(|\Sigma| + |L|))$
    - Bounded by complexity of identifying overlapping occurrences

Questions?