CMSC423: Chapter 9
Using suffix trees

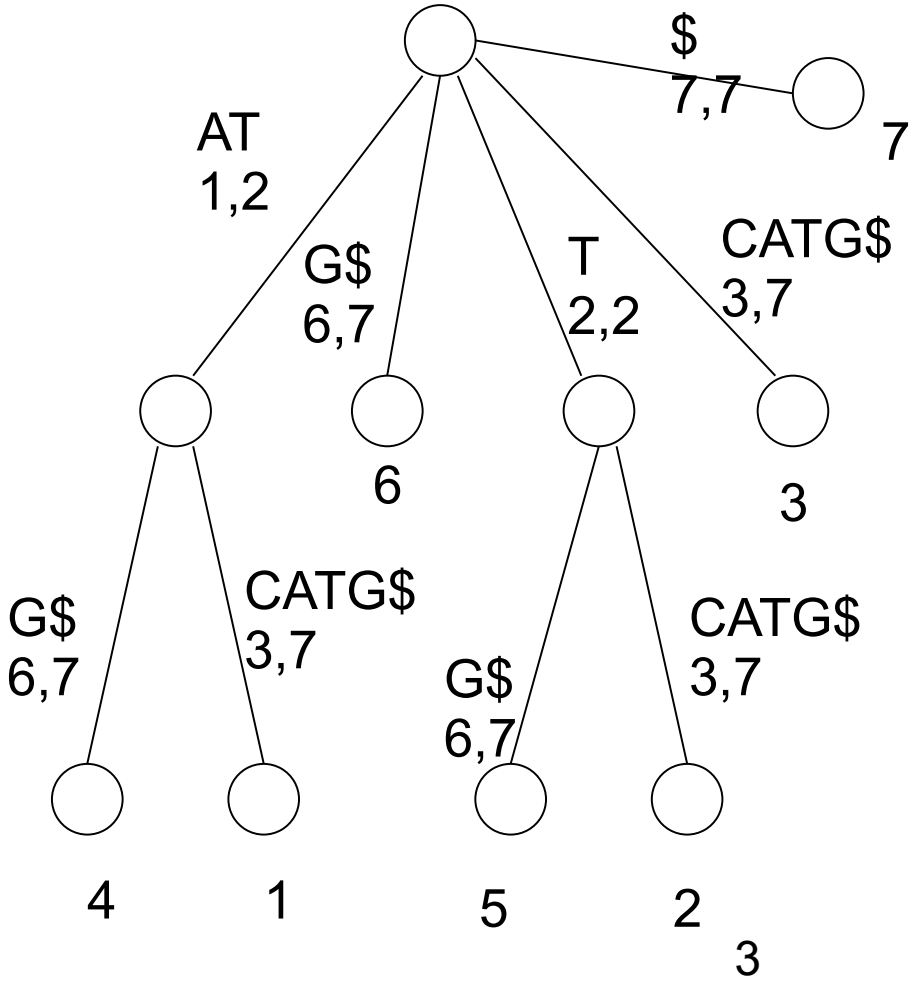
Suffix trees for matching

- Suffix trees use O(n) space
- Suffix trees can be constructed in O(n) time (trust me)
- Can we use them for matching?

Suffix trees for matching

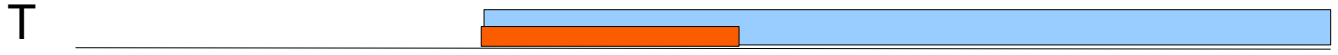
Is CAT part of ATCATG ?

- Match from root, char by char
- If run out of query – found match
- otherwise, there is no match



Intuition

- A match is the prefix of some suffix
- All strings with the same prefix follow the same path in a trie or suffix tree

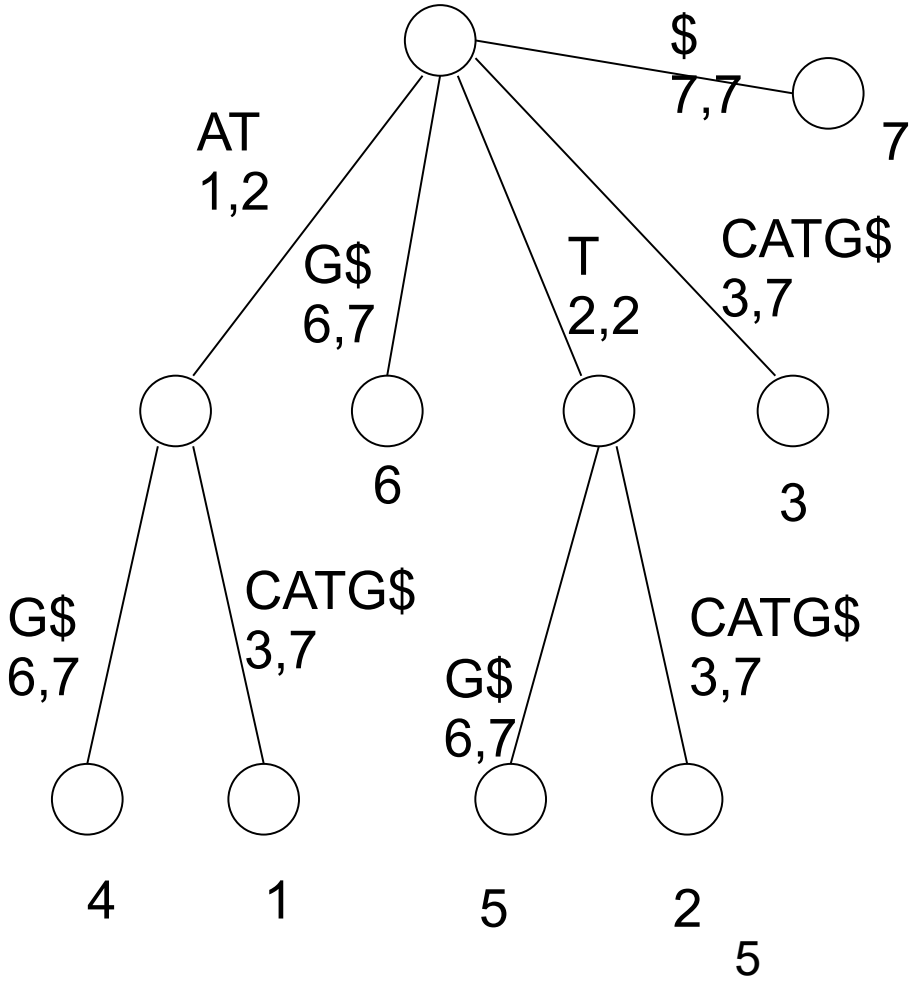


Suffix trees for matching

Is TCA part of ATCATG ?

- Match from root, char by char
- If run out of query – found match
- otherwise, there is no match

- What is the runtime?



Suffix trees for basic matching

- Finding if pattern is exactly found in text – O(len(pattern) + len(text))
- Same as KMP

  BUT!

  for k patterns of length m and text of length n

  KMP – O(k(m + n))

  Suffix trees – O(km + n)

Longest shared substring?

- Longest prefix match is trivial – "string depth" of the location where mismatch occurred
- Run longest prefix match for all suffixes of the pattern against suffix tree of text:

- $O(m^2 + n)$

- Using KMP it would be $O(m^2 + mn)$

- $O(m + n)$ possible, with some tricks

Other uses

- Finding repeats
  - Internal nodes with multiple children – DNA that occurs in multiple places in the genome
- Longest common substring of two strings
  - Build suffix tree from both strings. Find lowest (in terms of string depth) internal node that is the root of a sub-tree with leaves from both strings
- Shortest string not shared by two strings
  - Build suffix tree from both strings. Mark each node with the string composition of its subtree (does it have leaves from one or both of the strings). Assume they nodes are colored red (all leaves from the first string), blue (all leaves from the second string) , or purple (leaves from both strings). Find the highest (in terms of string depth) purple node that has a blue or red child.

Describe an algorithm that uses a suffix tree to compute Z values.

Write pseduo-code to compute the "string depth" of each node in a suffix tree.

Write pseudo-code to color nodes as done in the previous slide.

Describe an algorithm that computes the longest common prefix of two suffixes in a string.

Next: Other ways of managing suffixes