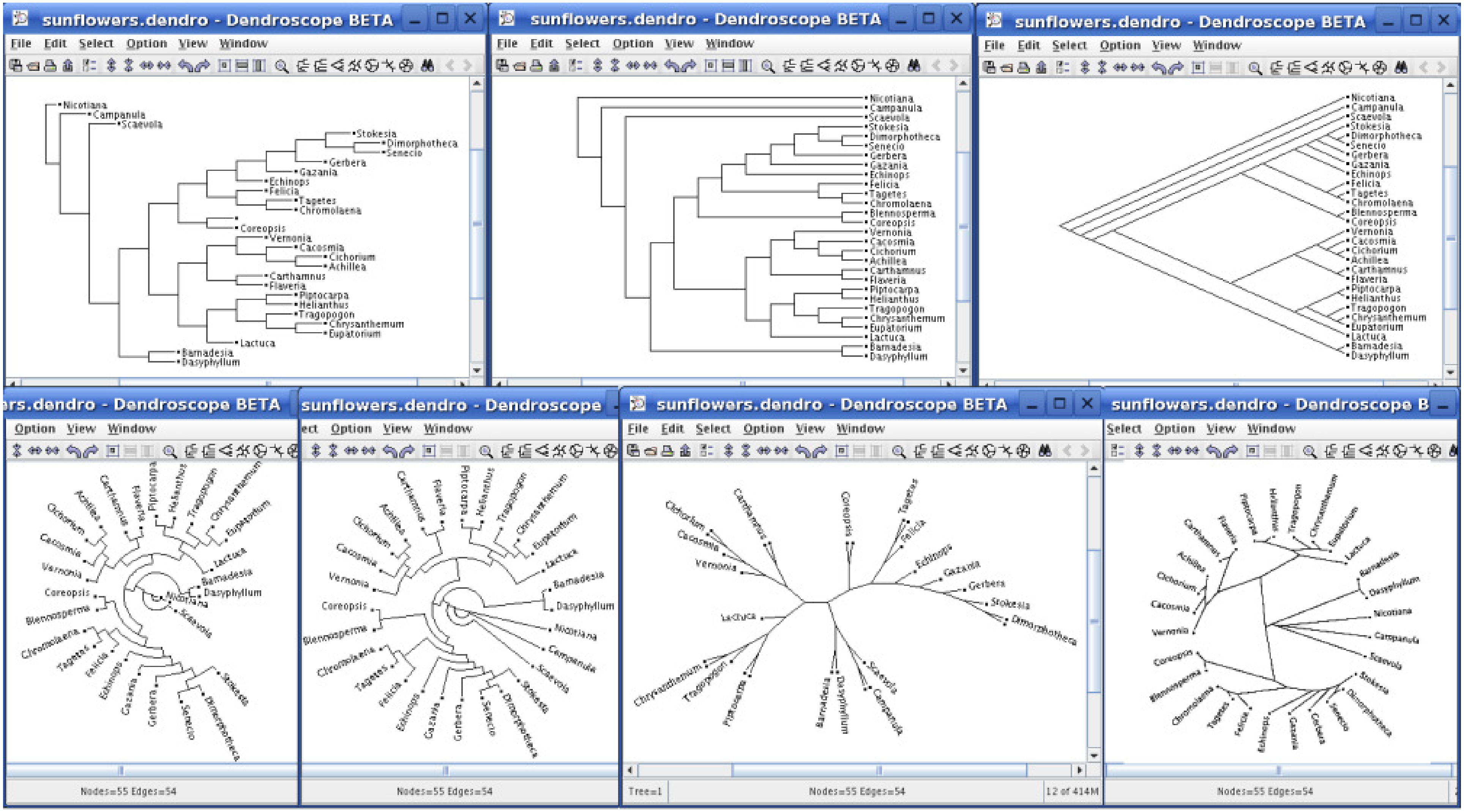
CMSC423: Bioinformatic Algorithms, Databases and Tools
Lecture 13

Phylogenetic tree display

Phylogenetic analysis

Suffix trees

Different tree views



http://www-ab.informatik.uni-tuebingen.de/software/dendroscope/welcome.html

Drawing trees

- Trees are easy to draw – just need to figure out how much space the leaves will take
- Step 1 – calculate how much space each node will take (how many leaves from current node)
- Step 2 – spread out the nodes according to # of leaves
- Many ways of optimizing: e.g. width, area

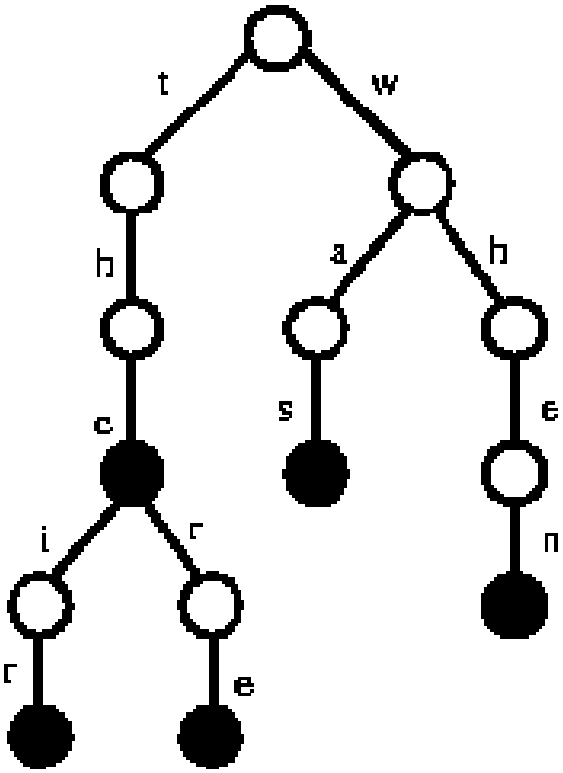
- For large trees
  - 3D displays (there's more room in 3D)
  - interactive displays (expand contract nodes as needed)

Analysis example

- Build multiple alignment (e.g. Muscle, ClustalW)
- Clean up alignment
  - manual editing
  - filters (pre-defined structure information)
- Build tree
  - PAUP – parsimony & others
  - Phylip – maximum likelihood
  - Tree-Puzzle –maximum likelihood
  - etc... (many packages)
- Integrated system – ARB
  - www.arb-home.de

Intro to suffix trees

- Used in fast exact matching
- Basic idea: extend a trie – structure for storing multiple strings

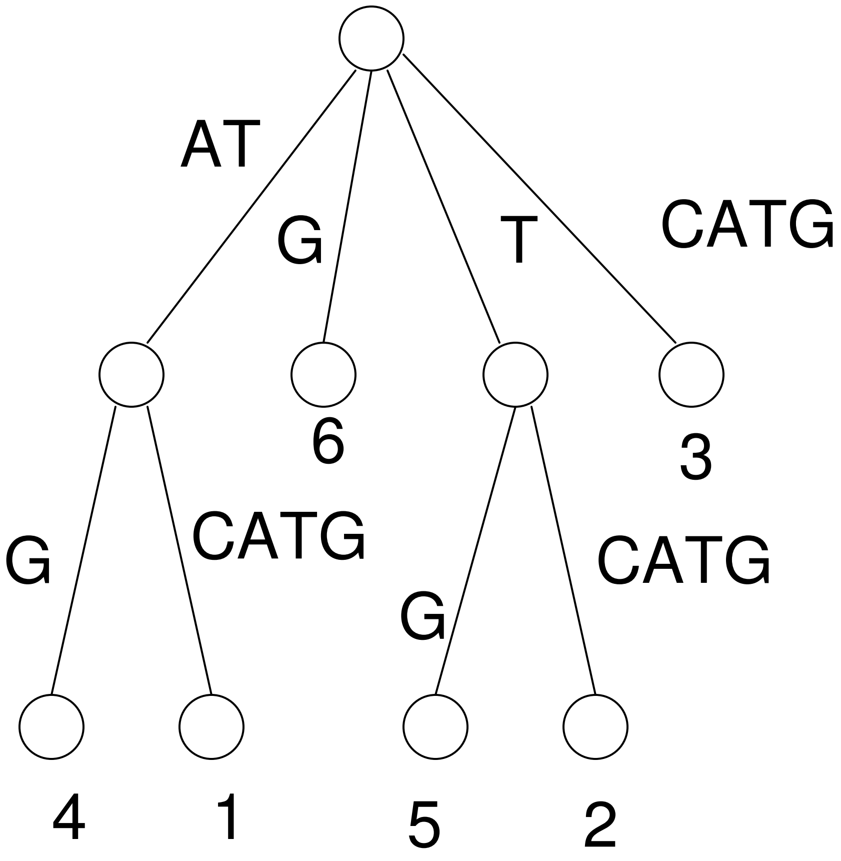


their
there
was
when

Suffix tree

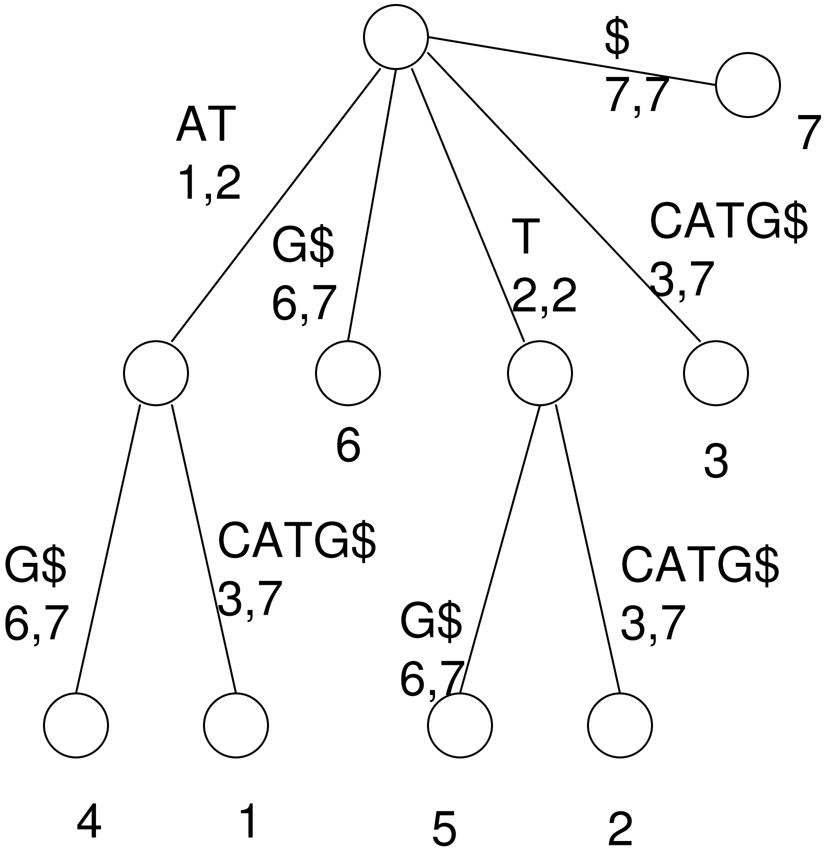
- Extends trie of all suffixes of a string

ATCATG
 TCATG
  CATG
   ATG
    TG
     G



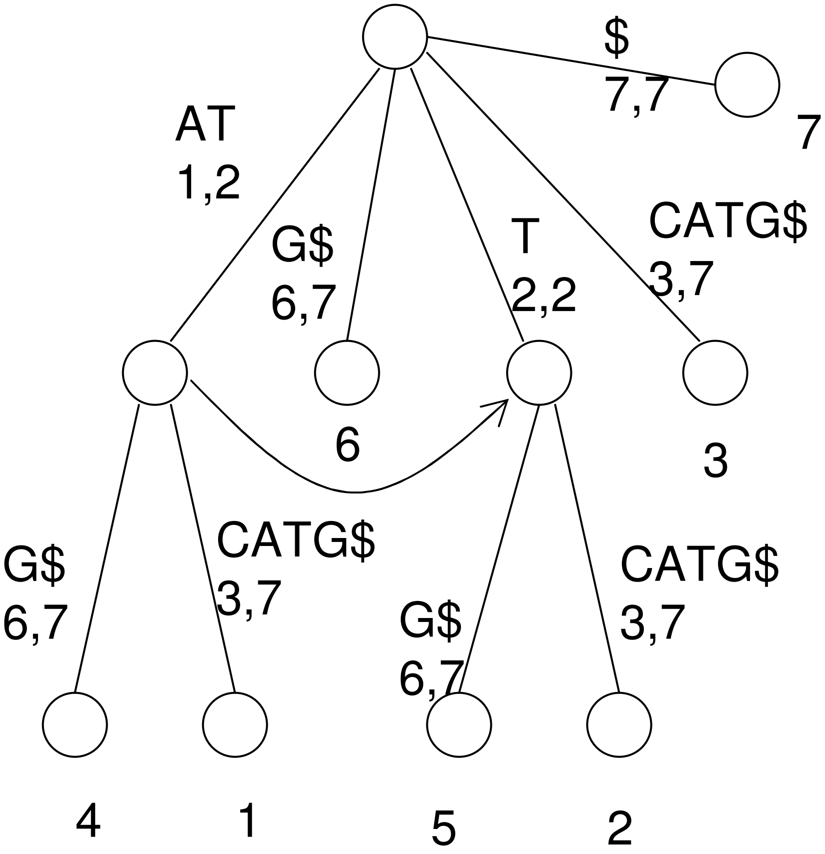
Suffix tree ...cont

- To store in linear time – just store range in sequence instead of string
- To ensure suffixes end at leaves, add $ char at end of string
- ATCATG$



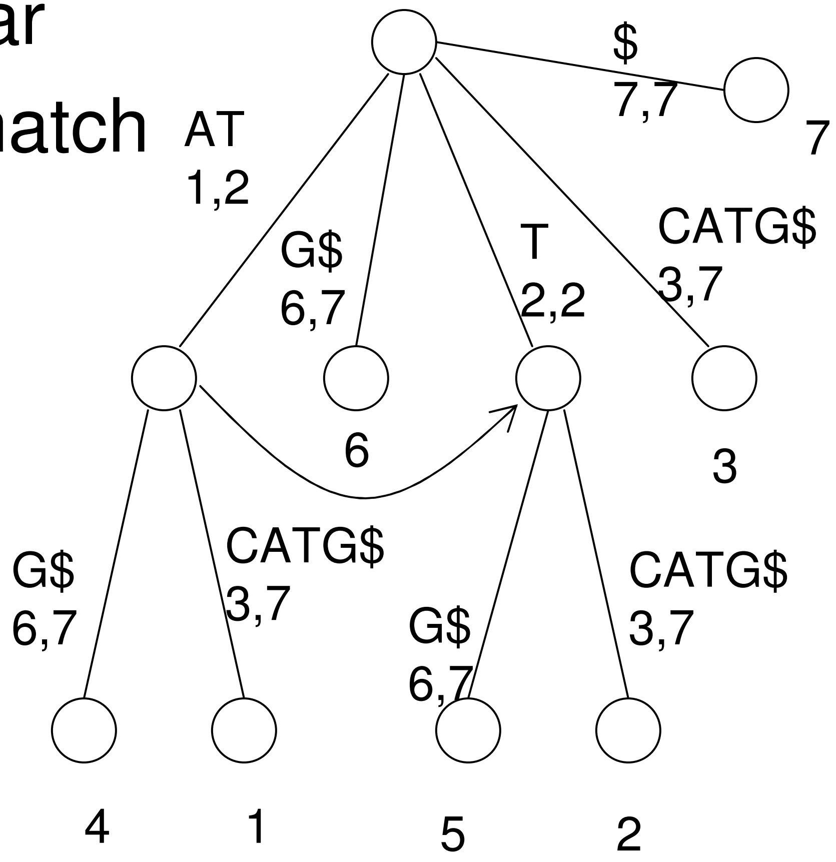
Suffix links

- Link every node labeled aS for some string S to node labeled S (note – it always exists)



Suffix trees for matching

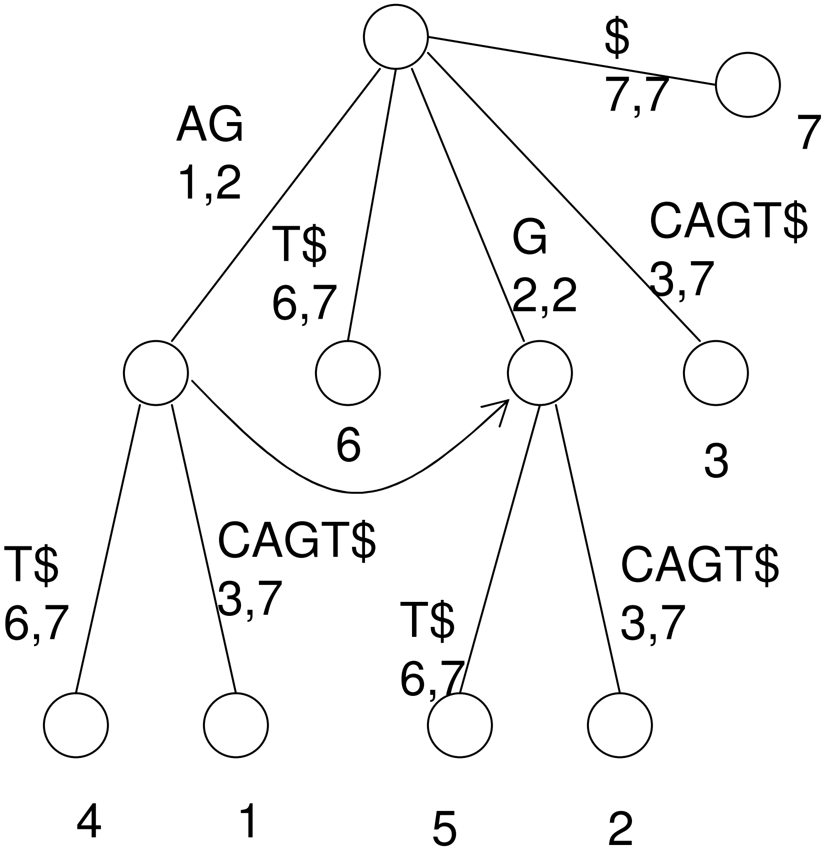
- Suffix trees use O(n) space
- Suffix trees can be constructed in O(n) time
- Is CAT part of ATCATG ?
- Match from root, char by char
- If run out of query – found match
- otherwise, there is no match



- intuition: CAT is the prefix of some suffix

Suffix links – useful for substring matches

- Does any part of AGATG match string AGCAGT?



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 of both strings. Find lowest internal node that has leaves from both strings

- Note: running time for matching is O(|Pattern|), not O(|Pattern| + |Text|)   (though O(|Text|) was spent in pre-processing

Suffix arrays

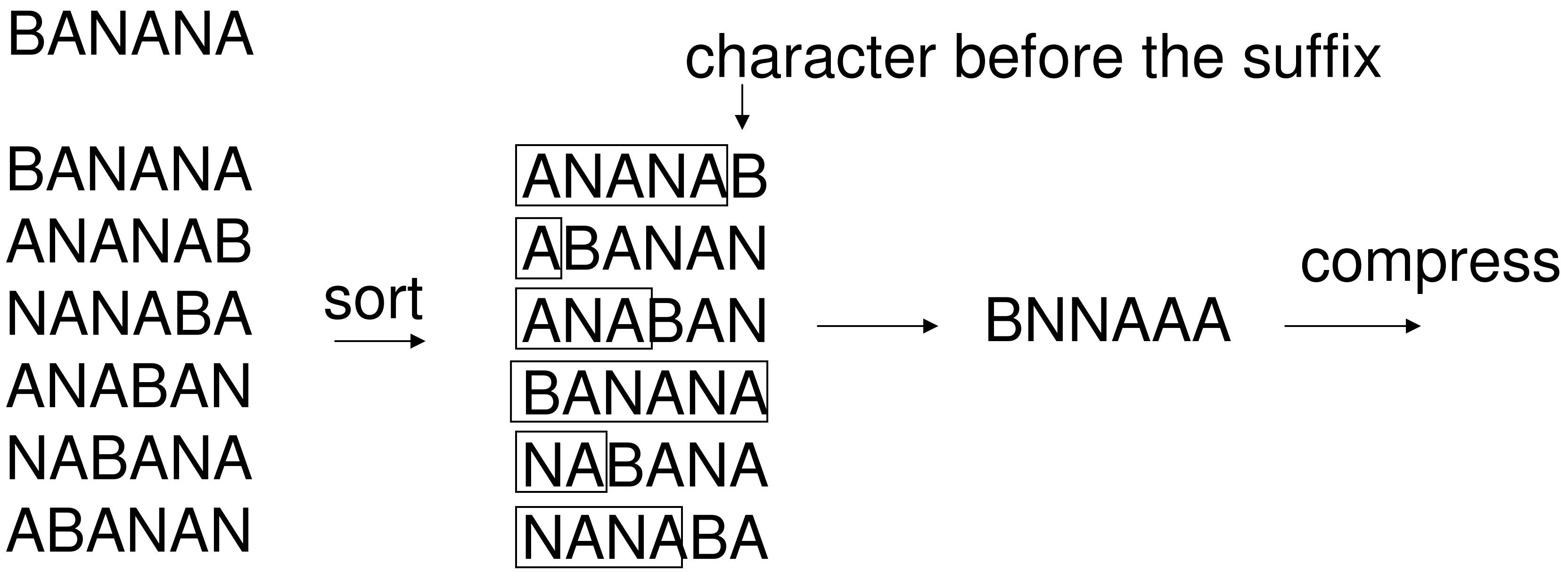
- Suffix trees are expensive > 20 bytes / base
- Suffix arrays: lexicographically sort all suffixes

```
   ATG
ATCATG
  CATG
     G
 TCATG
    TG
```

- Can quickly find the correct suffix through binary search
- Note: much less space, but longer running time (incur a log n term)

Suffix arrays and compression

- Burrows-Wheeler transform



BANANA

BANANA
ANANAB
NANABA
ANABAN
NABANA
ABANAN

sort

character before the suffix

ANANAB
ABANAN
ANABAN
BANANA
NABANA
NANABA

BNNAAA

compress