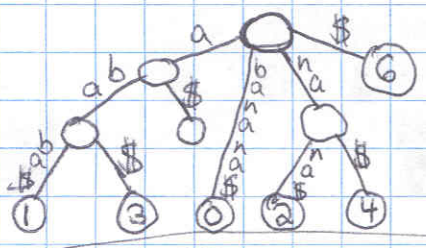
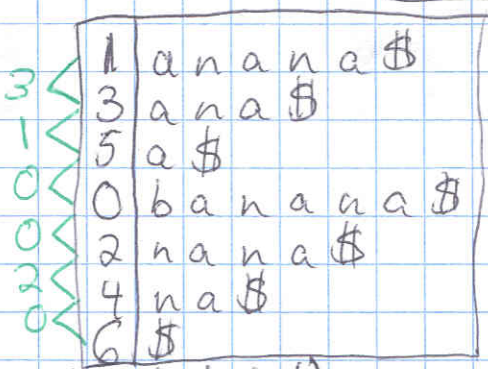
# Building suffix trees [Farach-FOCS1997; Farach-Colton, Ferragina, Muthukrishnan-JACM2000]

6.897
L11.2
Apr. 2, 2003

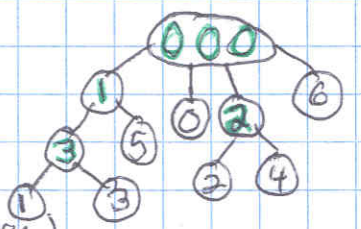



banana$
0123456




Suffix array:
- sort suffixes
- just store their indices
- can search in suffix array in $O(|P| + \lg|T|)$.
  - binary search, checking letters after what's already matched (on both L & R)

## Conversion
- stronger form of suffix array: store |lcp| between adjacent suffixes.
- suffix tree → suffix array
  - DFS, list leaves in order
  - compute lcp's using lca's (e.g.)
- suffix array → suffix tree
  - cartesian tree on lcp array
  - multiple equal min's all at root (nonbinary)
  - external nodes for suffixes in between lcp's.
  - internal nodes store "character depth" ⇒ can label edges




## Suffix links in suffix tree
- for any node labeled $ax$, there's a node labeled $x$.
  - any node is lca of 2 leaves, say labeled $T[i:]$ & $T[j:]$
  - ⇒ $\text{lcp}(T[i:], T[j:]) = ax$
  - ⇒ $\text{lcp}(T[i+1:], T[j+1:]) = x$
  - ⇒ lca of ~~leaves~~ labeled $T[i+1:]$ & $T[j+1:]$ is labeled $x$.
- can be computed in $O(1)$ time via lca
- suffix links form a tree, where depth(node) = character depth (length of label)
- so if you were given a suffix tree with leaf labels but no edge labels you could compute suffix-link tree, character depths, & now you know edges' character lengths ⇒ can compute edge labels