$ < a < ... < z

# Suffix Trees

Wednesday, March 3, 2021 10:55 AM

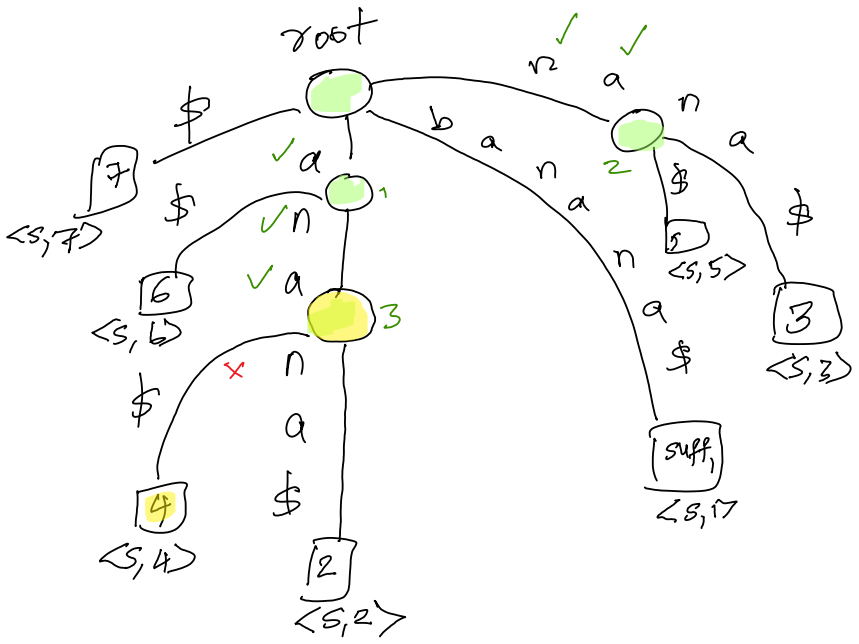
**Definition: Suffix tree:**

Given a string $S$ of length $n$, over alphabet $\Sigma$, the suffix tree of $S$ is the PATRICIA tree for all suffixes of $S$.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| S: | b | a | n | a | n | a | $ |

Suff#

1: b a n a n a $
2: a n a n a $
3: n a n a $
→4: a n a $
→5: n a $
6: a $
7: $




Claim: The path-label of every internal node is a repeating substring. (proof in Lemma 1)

ST: Suffix Tree:

ST properties:

- No. leaves = $n$
- No. internal nodes $\leq n-1$
- The path-label of leaf $i$ = suffix $i$ of $S$

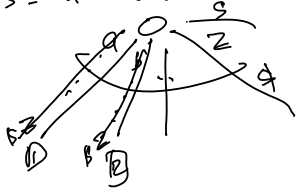
#edges = #nodes − 1

Space complexity = $O(n) + O(n^2)$ ← can be reduced to $O(n)$ if we store each edge label as an integer pair.

↑ for leaves & internal nodes

↑ for storing all edge labels.

(Think of a worst case example for this)

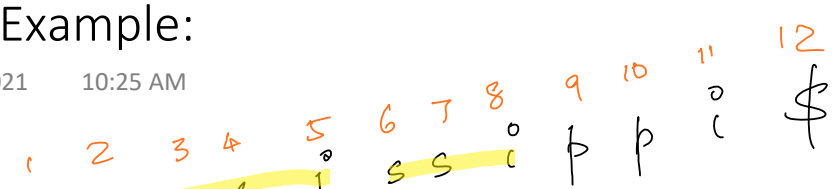



$n + (n-1) + (n-2) + \ldots 1 = O(n^2)$

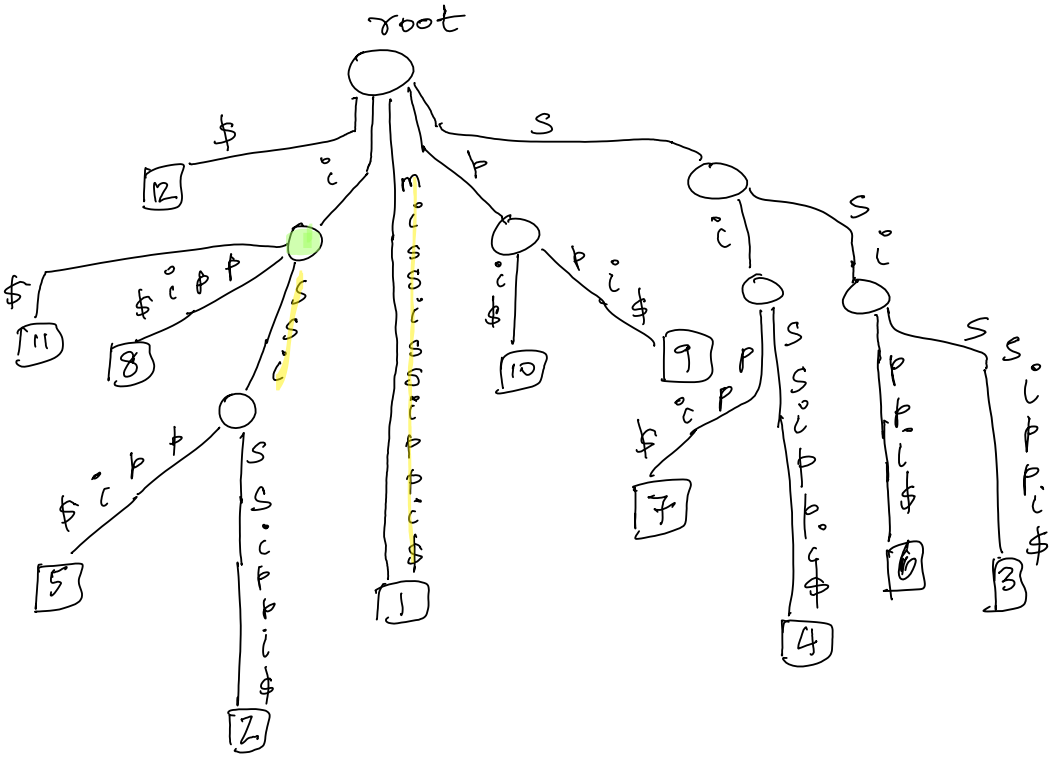
# Suffix Tree Example:

Thursday, March 4, 2021 10:25 AM

Example: S = m i s s i s s i p p i $



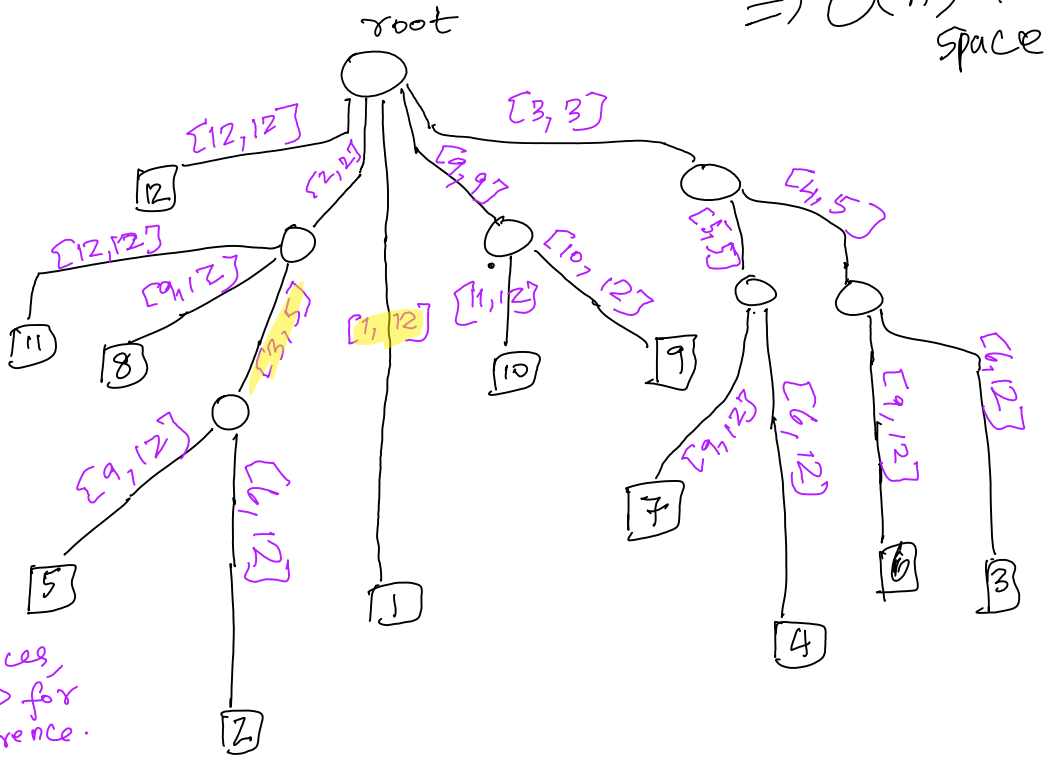

ST(S):




Observation:
Every edge label is a substring in S.

=> O(n) tree space

ST(S):
with **edge label compaction**




- Every edge label is stored as an integer pair <start index, end index> where it occurs as a substring in S.
- If label occurs as a substring in multiple places, just store <start, end> for any occurrence.

# Suffix Tree Implementation

Thursday, March 4, 2021 2:40 PM

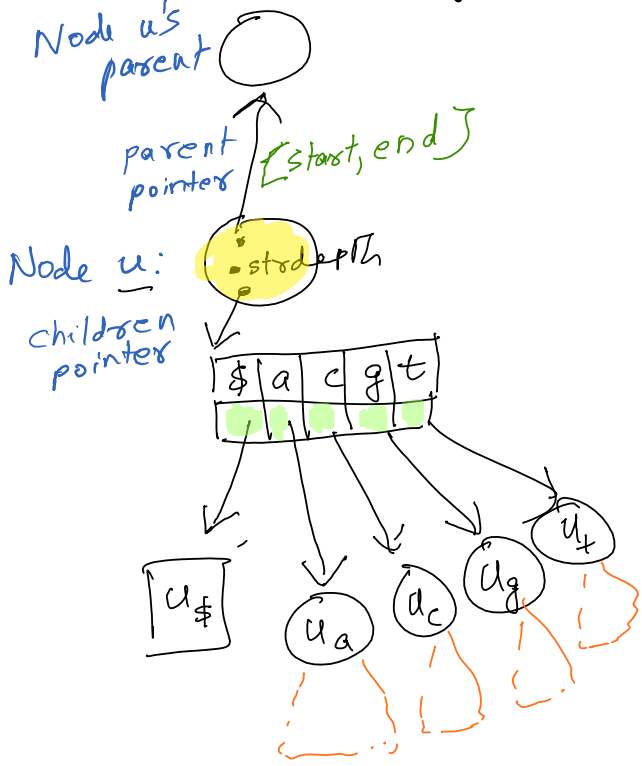
Q) How to implement the tree data structure (in a space efficient way)?

## Node structure:

```
struct/class Node {
    int id;                          // 4 bytes
    int string_depth;                // 4
    struct Node *parent;             // 8
    pair<int, int> <start, end>;     // 8
    struct Node *child[1≤|+|];       // 40
}
```

64 bytes per node

#nodes < 2n

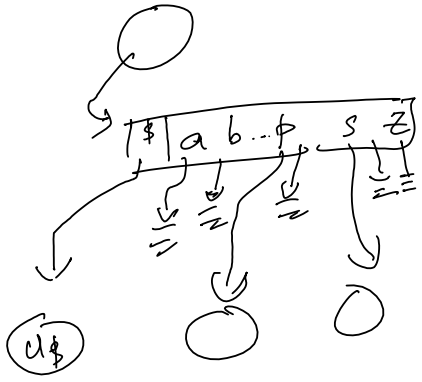



$n = 10^6$

$2n \times 64 = 128n$

space constant

| Input string size | |
|---|---|
| $n = 10^6$ | $\Rightarrow$ 128MB free space |
| $10^9$ | $\Rightarrow$ 128 GB free space |



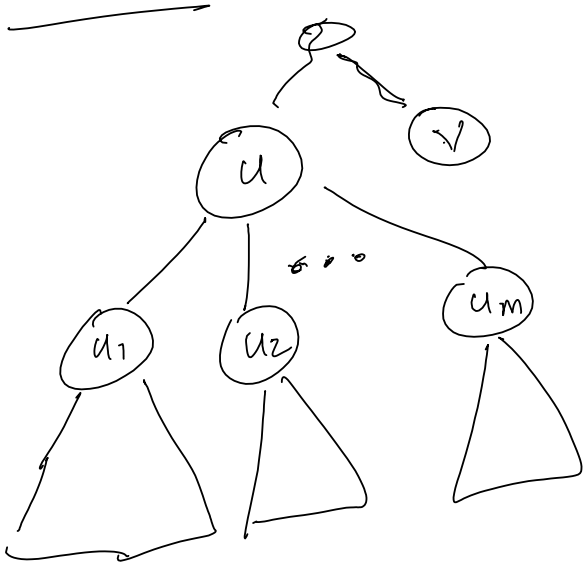
# Space efficient Tree Implementation Tricks

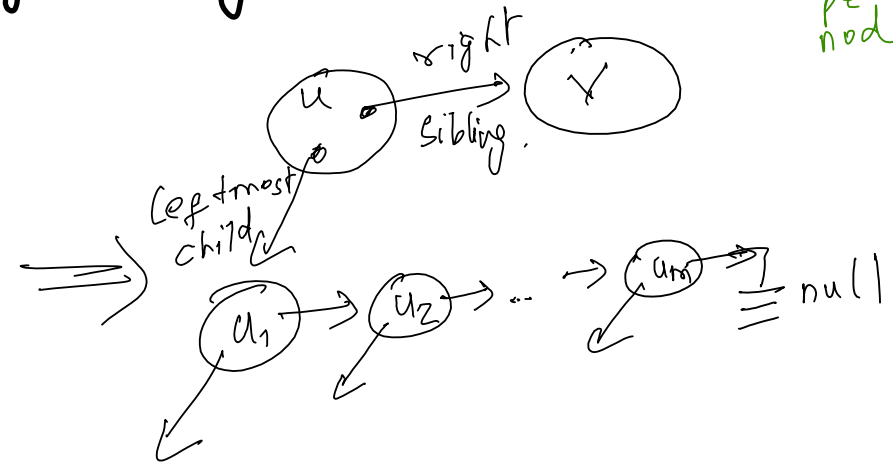
Thursday, March 4, 2021 10:45 AM

Given an arbitrary m-way tree T, there are a couple of ways to reduce the space consumption. (memory footprint).

## Trick #1 left-child, right-sibling approach

2 pointers per node



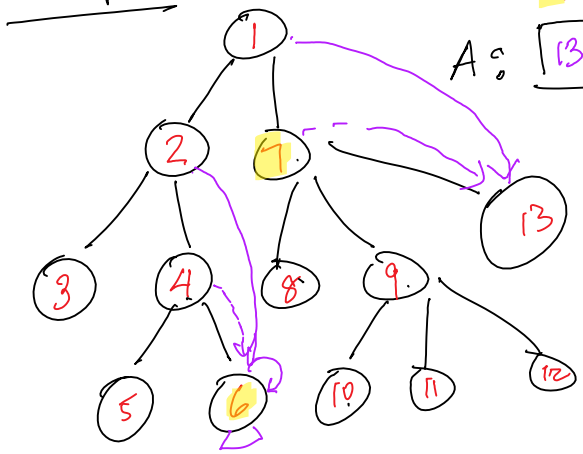




1 pointer per node

## Trick #2 The Rightmost Leaf Approach:

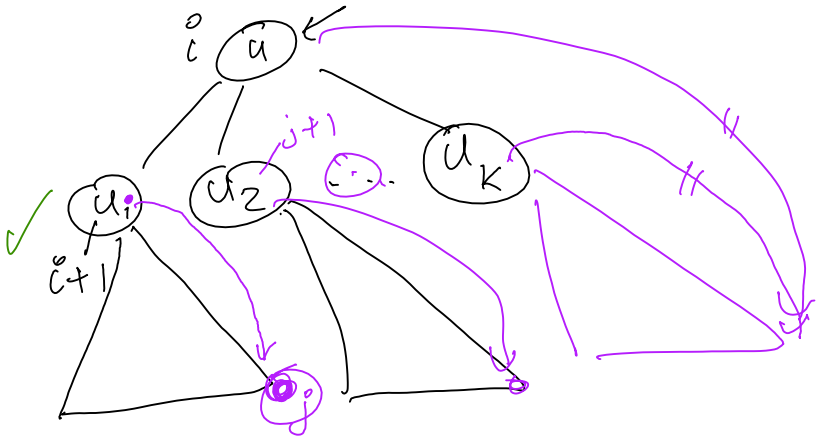
1. store nodes in DFS order
2. Each node stores only one pointer (the pointer to the rightmost leaf under its subtree).

Example:




A:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 6 | 3 | 6 | 5 | 6 | 13 | 8 | 12 | 10 | 11 | 12 | 13 |



Definition: A "repeat" of a string $s$ is a substring that occurs atleast twice in $s$.

# The Longest Repeat Problem

Wednesday, March 3, 2021 11:05 AM

Q1) Given $S[1..n]$, find a longest repeating substring ("repeat")

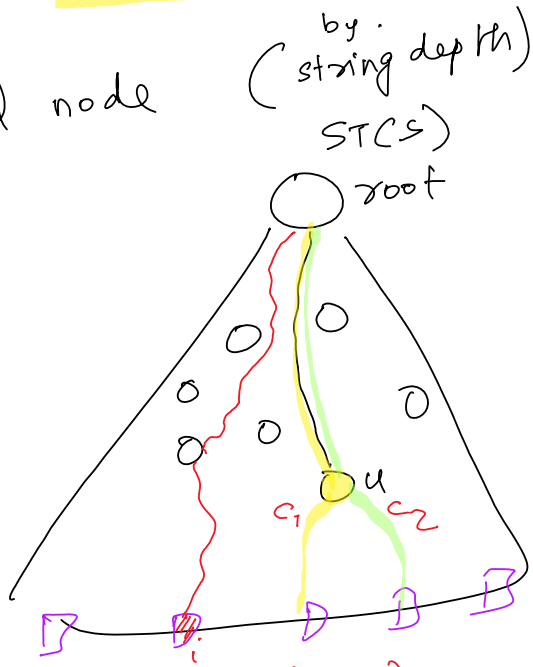
b a n a n a ⇒ "ana"

Algo:

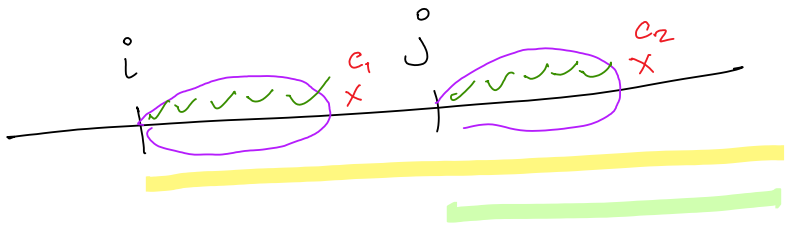
1) Build ST(S)

2) Look for any "deepest" internal node (by string depth)
→ call this node $u$

3) Output pathlabel(u)




pathlabel(leaf i) = suff$_i$




Claim: The algorithm above will output a longest repeat of $S$ always (i.e., will not miss it).

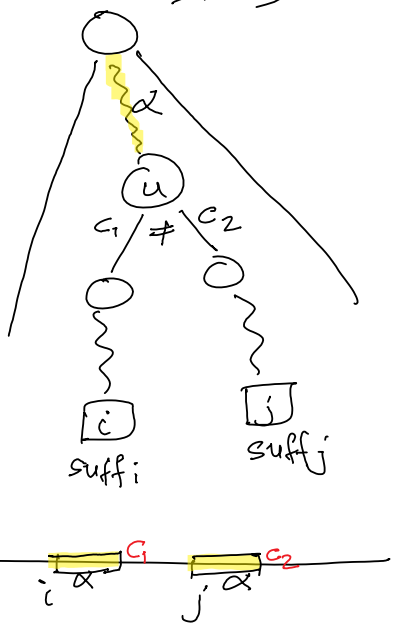
To prove this claim, we will prove it in two parts

(see next page)

# Internal Nodes and Repeating Substrings

Thursday, March 4, 2021 9:51 AM

**Lemma 1.** Let $u$ be any internal node in $ST(S)$. Then the path-label($u$) is a repeat in $S$.







**Proof:**

Since $u$ is an internal node

$\Rightarrow$ $u$ has at least two children

$\Rightarrow$ those two children lead to at least two leaves, say $i$ & $j$.

$\Rightarrow$ Since pathlabel(leaf$i$) = suff$i$ and pathlabel(leaf$j$) = suff$j$ :

$\Rightarrow$ suff$i$ and suff$j$ share ($\alpha$ = pathlabel($u$)) as a common prefix

$\Rightarrow$ $\alpha$ is the lcp(suff$i$, suff$j$) because $c_1 \neq c_2$ (right maximality)

$\Rightarrow$ And, since $i \neq j$,

$\alpha$ has to be a (right-maximal) repeat in $S$. □

**Corollary:** If $u$ is a deepest internal node in $ST(S)$, it implies that $\alpha$ (= pathlabel($u$)) is also a longest repeat in $S$. □