Adaptive Huffman Code

Sudeep Narala
Raymond Yang

Overview

- Huffman codes require two passes through data: construct tree then encode
- Adaptive huffman codes require only one pass, building the code while transmitting the symbol

Optimality of Vitter’s

- Sibling Property
  - Every node (except root!) has a sibling
  - As you go from bottom to top, left to right in tree, node weights are non-decreasing
  - Internal nodes and leaves weighted just like Huffman tree
- Any tree that maintains sibling property is a valid Huffman tree for that distribution
- Vitter’s Algorithm constructs the optimal tree by maintaining sibling property invariant

Encode Symbol Pseudocode

```
algorithm for adding a symbol is
    leaf_to_increment := NULL
    p := pointer to the leaf node containing the next symbol

    if (p is NYT) then
        Extend p by adding two children
        Left child becomes new NYT and right child is the new symbol leaf node
        p := parent of new symbol leaf node
        leaf_to_increment := Right Child of p
    else
        Swap p with leader of its block
        if (new p is sibling to NYT) then
            leaf_to_increment := p
            p := parent of p

    while (p ≠ NULL) do
        Slide_And_Increment(p)

    if (leaf_to_increment != NULL) then
        Slide_And_Increment(leaf_to_increment)
```

Encode Symbol Pseudocode

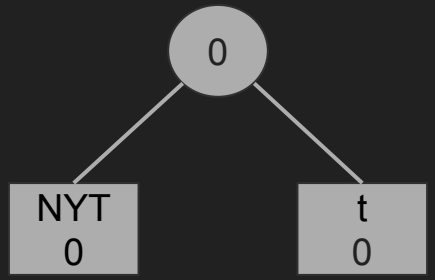
```
function Slide_And_Increment(p) is
    previous_p := parent of p

    if (p is an internal node) then
        Slide p in the tree higher than the leaf nodes of weight wt + 1
        increase weight of p by 1
        p := previous_p
    else
        Slide p in the tree higher than the internal nodes of weight wt
        increase weight of p by 1
        p := new parent of p.
```

Example: “test”

Initialization

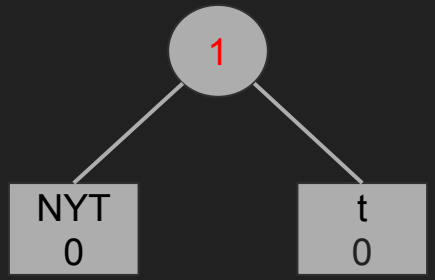
Example: “test”

Input: “t”



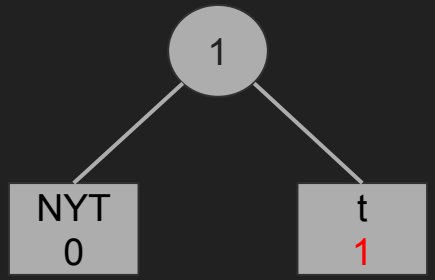
Example: “test”

Input: “t”



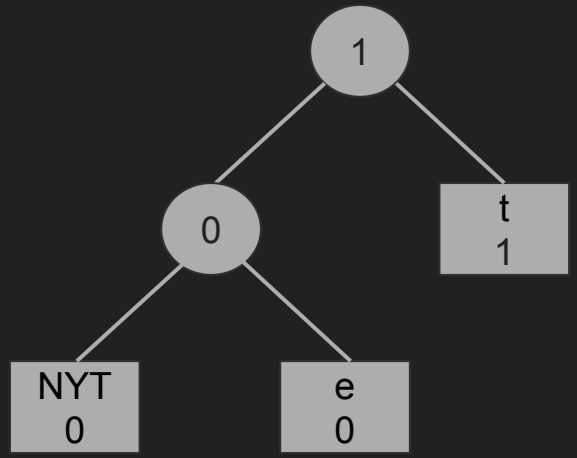
Example: “test”

Input: “t”



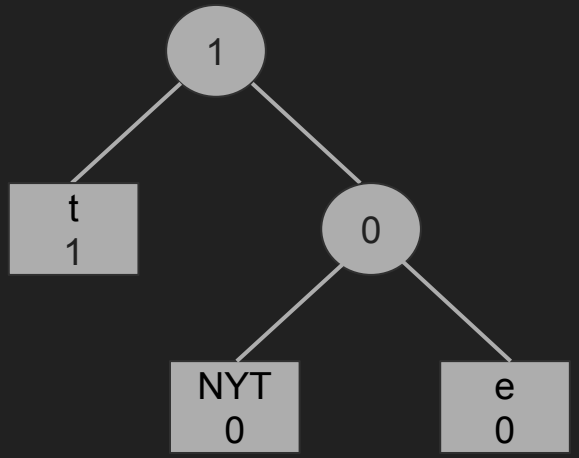
Example: “test”

Input: “e”



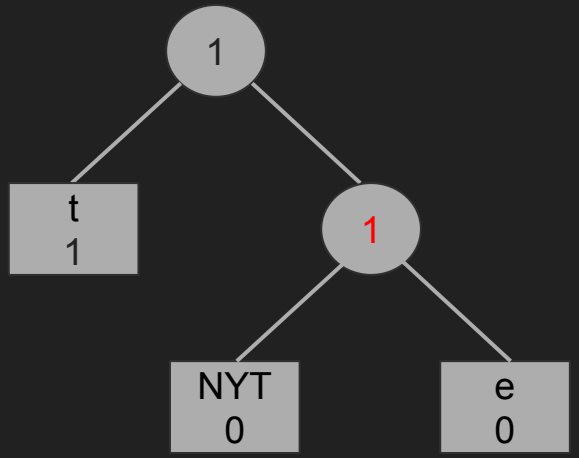
Example: “test”

Input: “e”



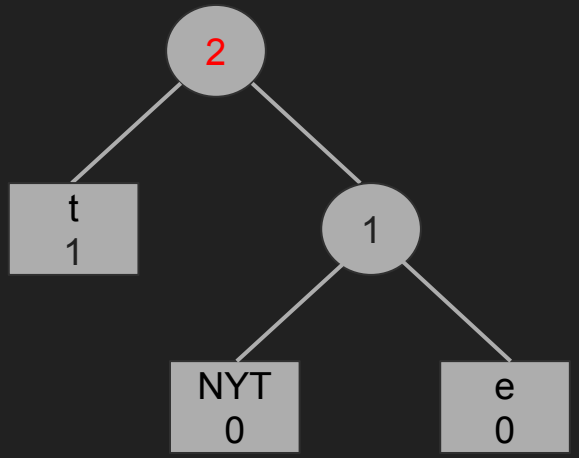
Example: “test”

Input: “e”



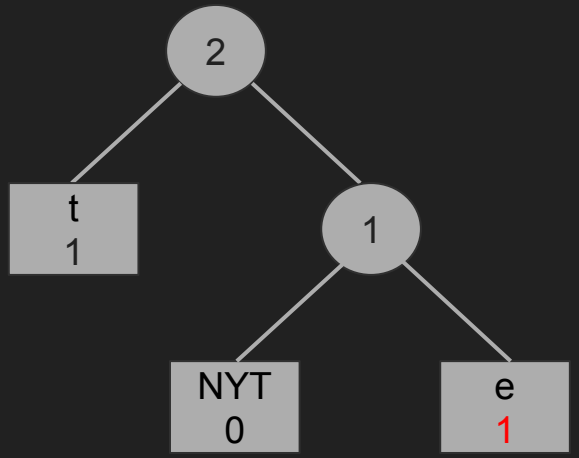
Example: “test”

Input: “e”



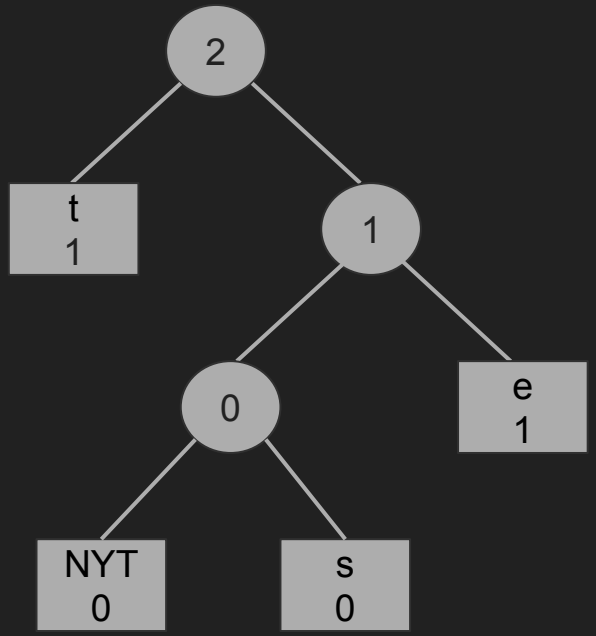
Example: “test”

Input: “e”



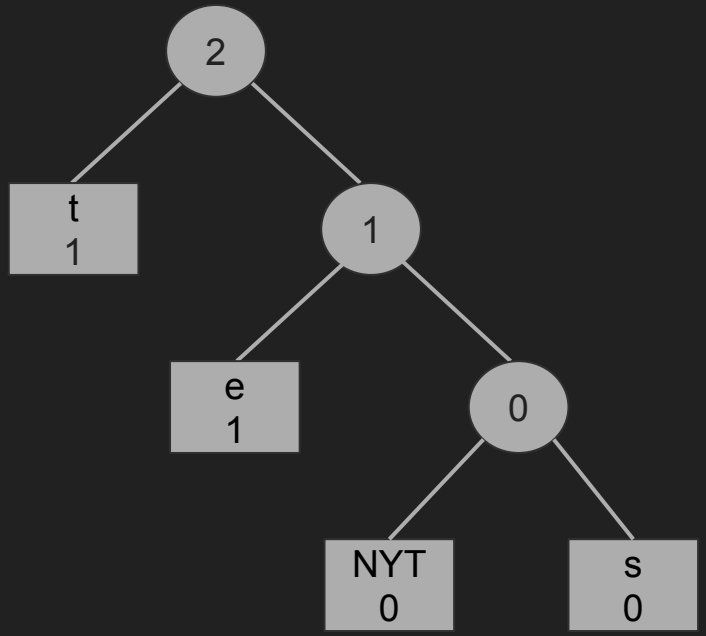
Example: “test”

Input: “s”



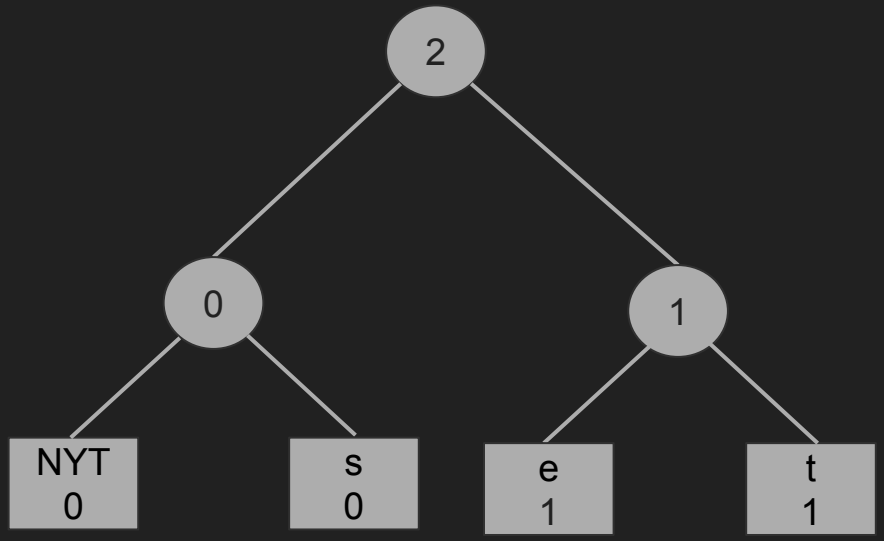
Example: “test”

Input: “s”



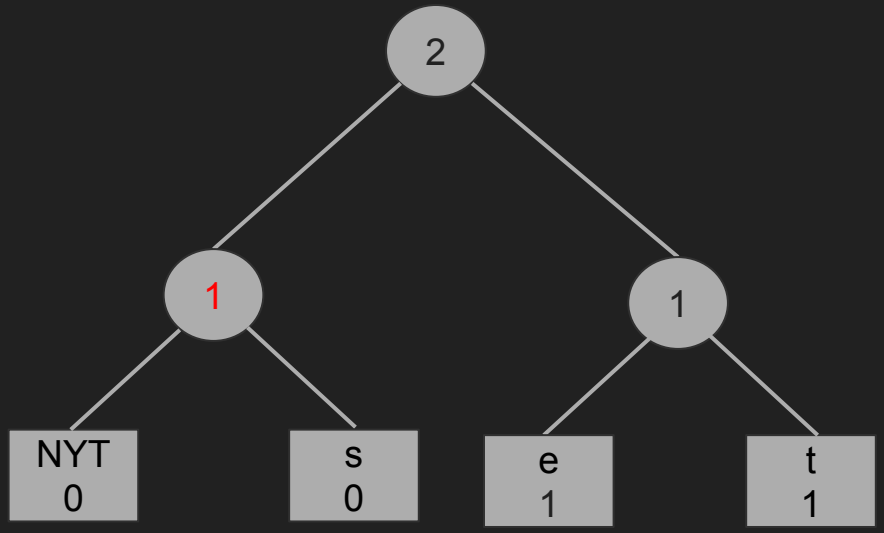
Example: “test”

Input: “s”



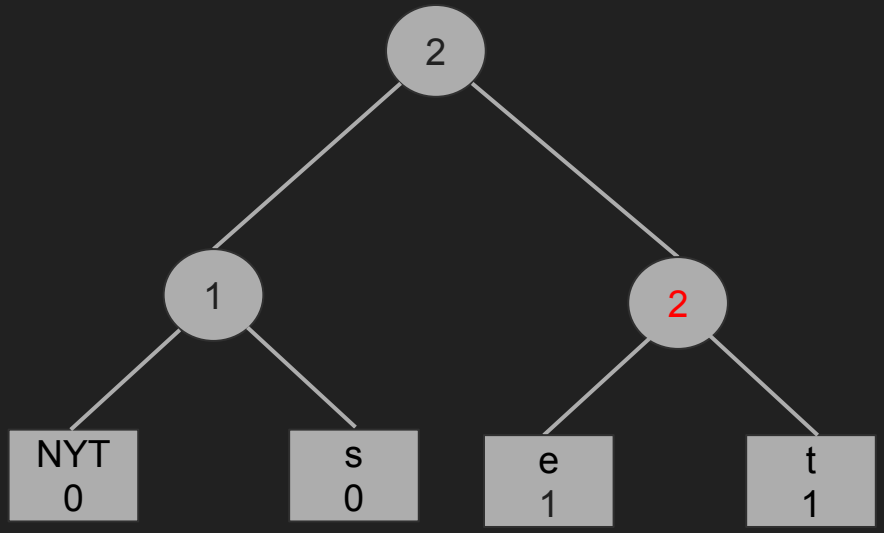
Example: “test”

Input: “s”



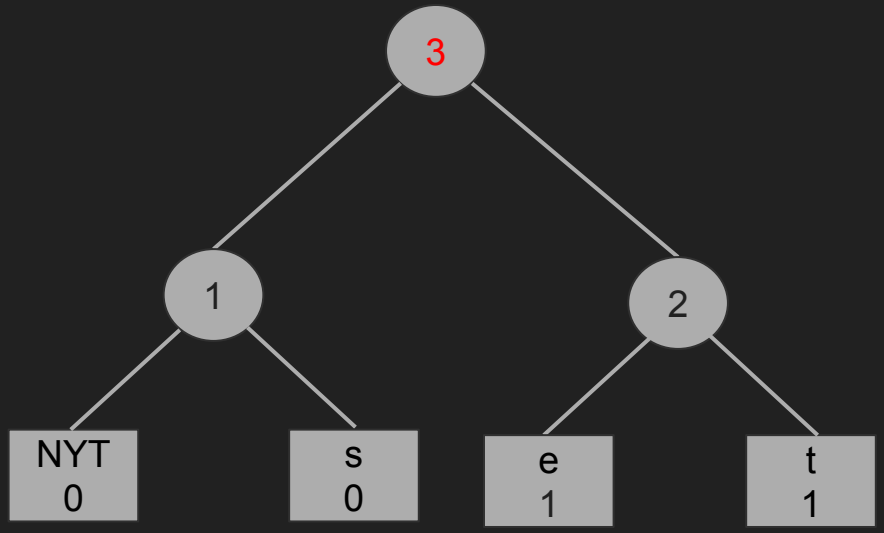
Example: “test”

Input: “s”



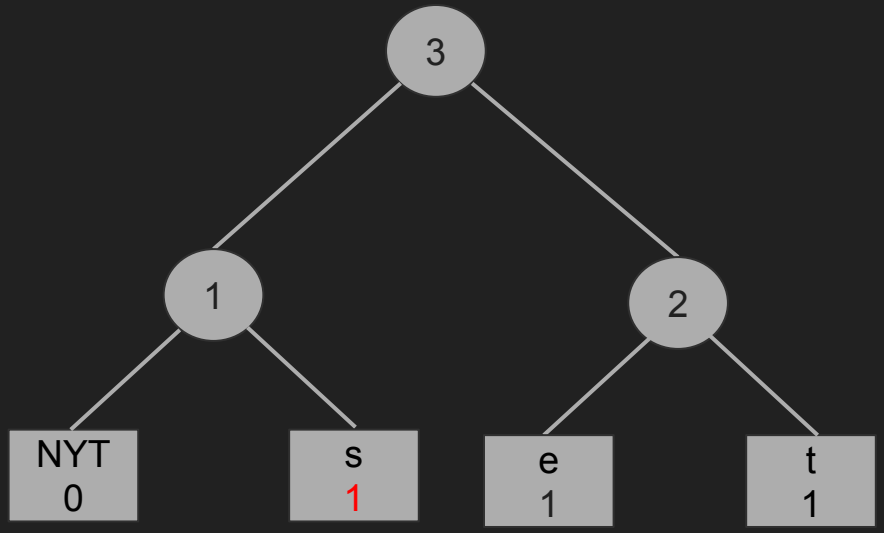
Example: “test”

Input: “s”



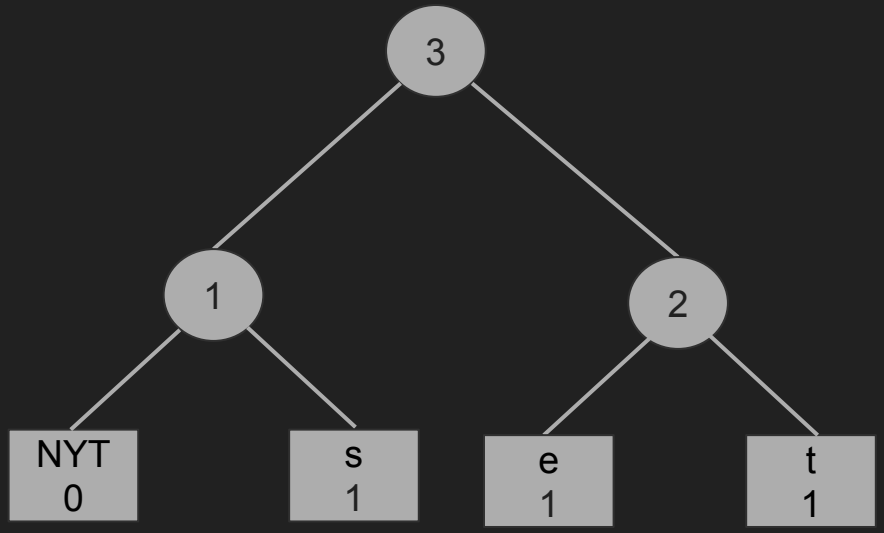
Example: “test”

Input: “s”



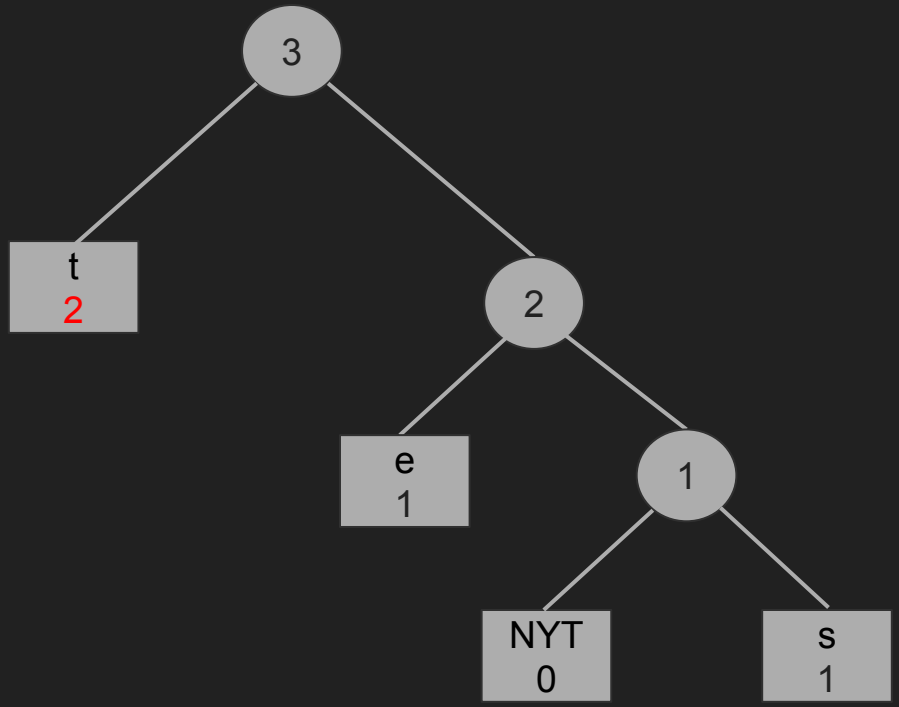
Example: “test”

Input: “t”



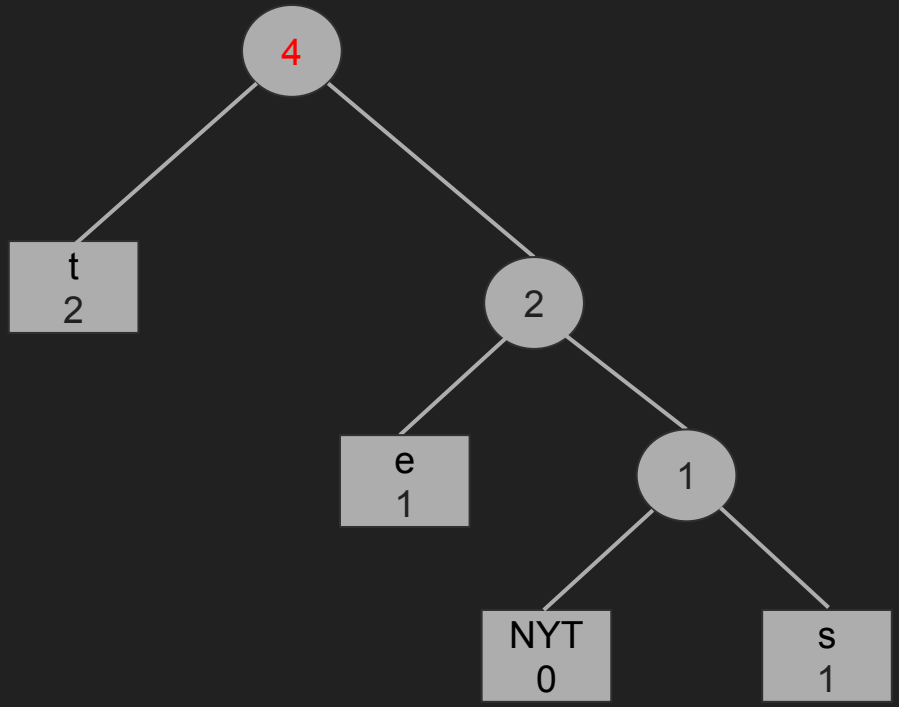
Example: “test”

Input: “t”



Example: “test”

Input: “t”



Questions?