a
1 2 3
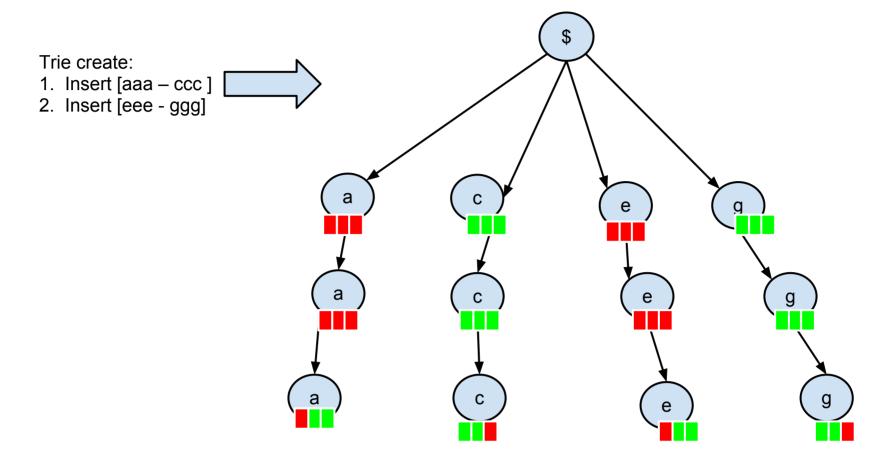
A single TrieNode with pointer
from parent node and pointers to
child nodes [0-127]. Three flags
represent the following:
1. LeftSide is valid.
2. Node and Subtrie is valid.
3. RightSide is valid.
Red is invalid. Green is valid.

Trie create:
1. Insert [aaa – ccc ]
2. Insert [eee - ggg]

$

a        c        e        g

a        c        e        g

a        c        e        g

TestCase1 ( non-overlapping ranges)
i. Search for bbb
ii.  Search for ddd
TestCase2 ( overlapping ranges)
iii. Search ddd
iv.  Search bbb

Pseudocode:
a. start at root node $. node = root
b. for character,c in input:
        if c is not childnode of node
            prev_node = node.predecessor(...)
            next_node = node.sucessor(...)
            return true, if prev.right_valid and next.left_valid
        else
            node = childnode
c. If node.valid: //Valid eaf node for input in trie
        return true

$

a        bbb        c

a        c

a        c

i) 'bbb' is not found in trie.
Predecessor = 'aaa'
Successor = 'c'
Predeccesor.right == Sucessor.left = true.
Return true.