```java
import java.util.*;

public class Trie2 {

    private class TrieNode {
        Map<Character, TrieNode> children = new TreeMap<>();//
TreeMap is java build-in structure,
        boolean aword = false;                              //
Basically it acts like a Hashtable or Hashmap, establishing a
mapping between Key and Value
                                                     //Unlike
hash table, keys in TreeMap are sorted!
    }

    private TrieNode root;
    public Trie2() {
        this.root = new TrieNode();
    }

    public void insertString(String s) {
        insertString(root, s);
    }

    private void insertString(TrieNode root, String s) {
        TrieNode cur = root;
        for (char ch : s.toCharArray()) {
            TrieNode next = cur.children.get(ch);
            if (next == null)
                cur.children.put(ch, next = new TrieNode());
            cur = next;
        }
        cur.aword = true;
    }
}
```
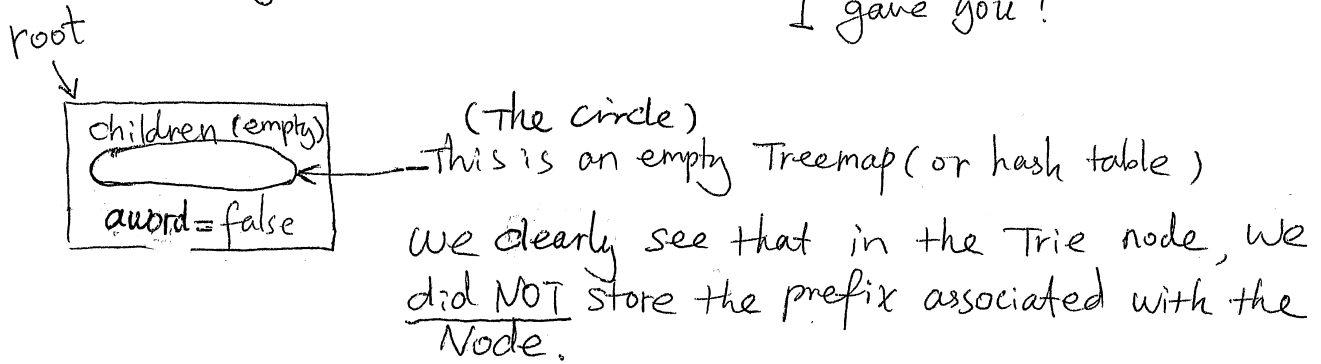
① Construct an empty Trie (prefix tree) with the implementation
I gave you!

root

children (empty)

aword = false

(The circle)
This is an empty Treemap (or hash table)

We clearly see that in the Trie node, we
did NOT store the prefix associated with the
Node.

②
insert string into Trie; using insertString method:

insert word "hello" (The first word we insert)

initially: cur = root;

for Loop 1st iteration:
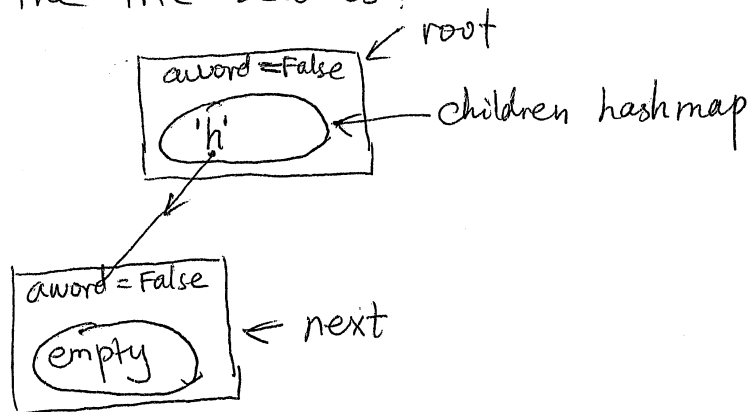ch = 'h'
next = null, ⟺ cur.children.get(ch) // here, checking
whether there is a child that letter 'h' associated with.
if no such child (ie. next is null)

We make a new TrieNode to be associated with
letter 'h', which is one of children of cur.

The Trie becomes:

aword = False

'h'

← root

children hashmap

aword = False

empty

← next

Then move cur; ⇒ cur = next;

①

insert string "hello" be continued, (insert First word into Trie)
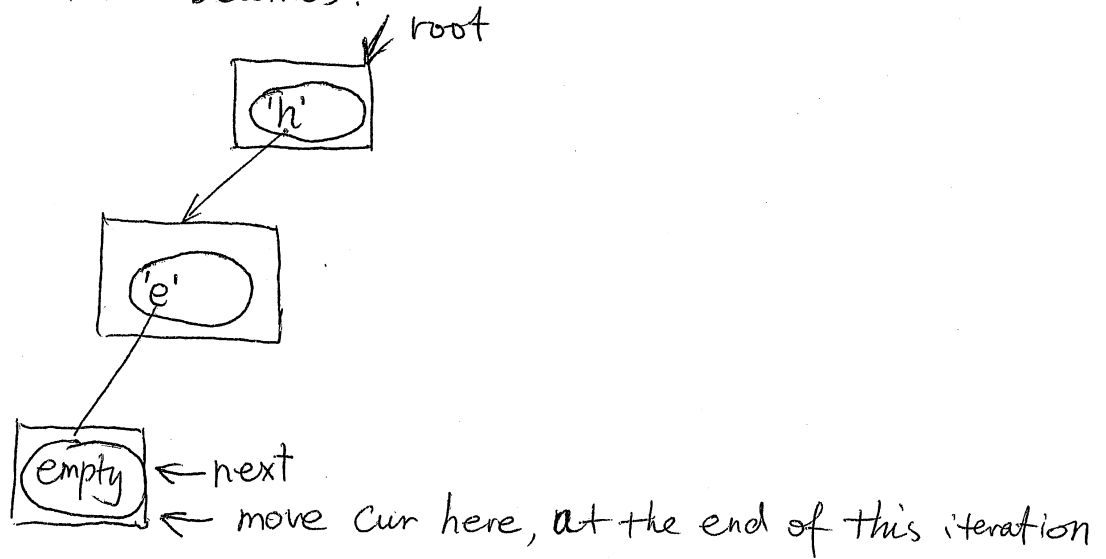
For Loop 2nd iteration:
    ch = 'e', cur → the empty new node, created
    next = cur. children.get('e') in previous iteration.
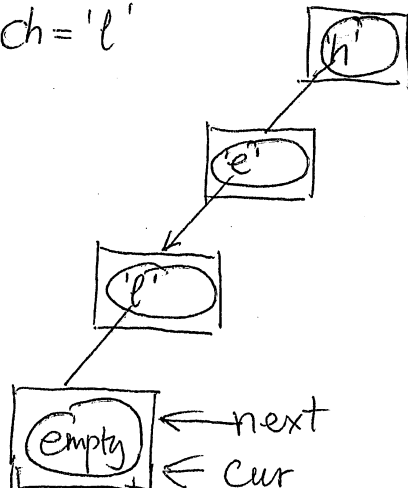   next gets null; because No such child;
    if so,
        We make a new TrieNode to be associated with letter 'e'
        in Current node, The new Node is a child of cur.
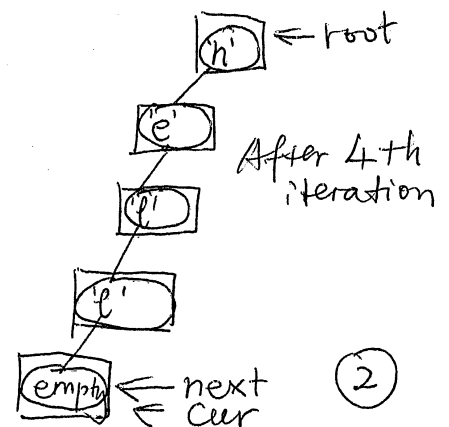The Trie becomes:

For Loop 3rd iteration:
    ch = 'l'

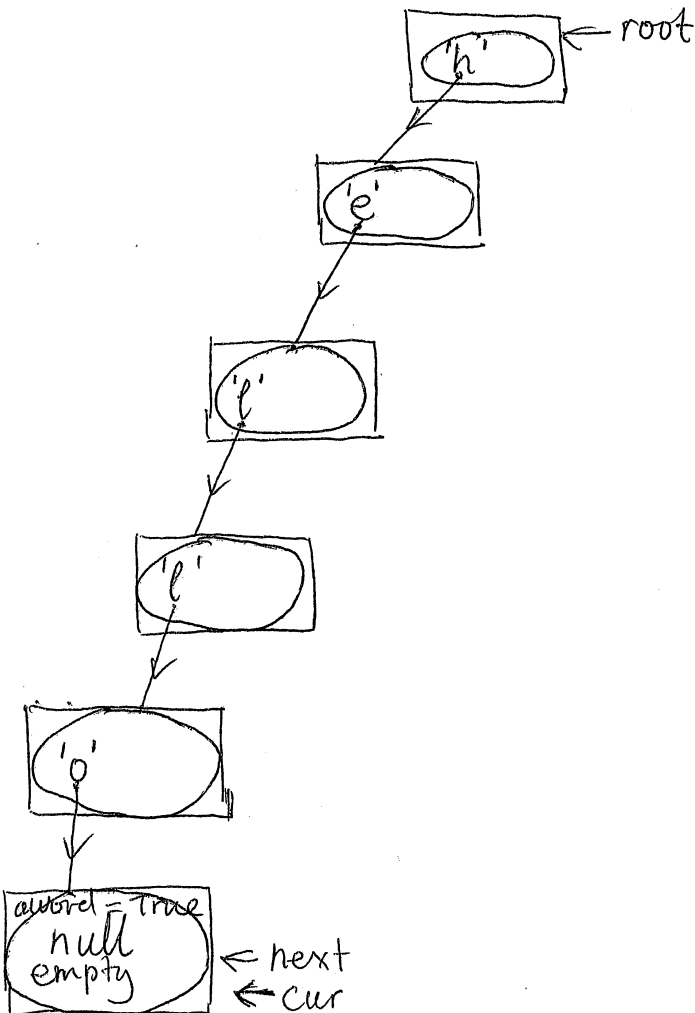For Loop 4th iteration: Skipped

    5th iteration on next page

②

inserting String "hello" (The First word be inserted)

For Loop 5th iteration:

ch = 'o' the Last Letter in word;

We find no child in cur that is associated with letter 'o'

we create a new Node For this Letter

the after 5th iteration, For drops out,

Assign true to cur.aword

Two Questions To think!

① Use the Current Trie above, what happens if we Continue inserting 'he'

② Use the Current Trie above, what happens if we Continue inserting 'her'

③