648. Replace Words

```
Input: dictionary = ["cat","bat","rat"], sentence = "the cattle
was rattled by the battery"
Output: "the cat was rat by the bat"
```

cattle :

c

ca

cat

catt

cattl

cattle

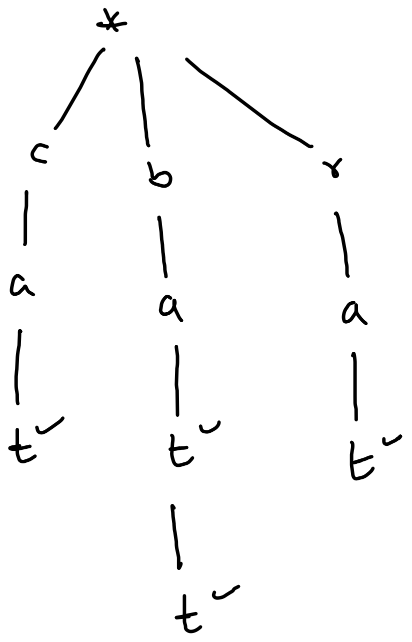
cat rat bat

the ~~cattle~~ was ~~rattled~~ by the ~~battery~~

batt

```
Input: dictionary = ["cat","bat","rat"], sentence = "the cattle
was rattled by the battery"
Output: "the cat was rat by the bat"
```

trie :



cat rat

the ~~cattle~~ was ~~rattled~~ by the

~~battery.~~

bat

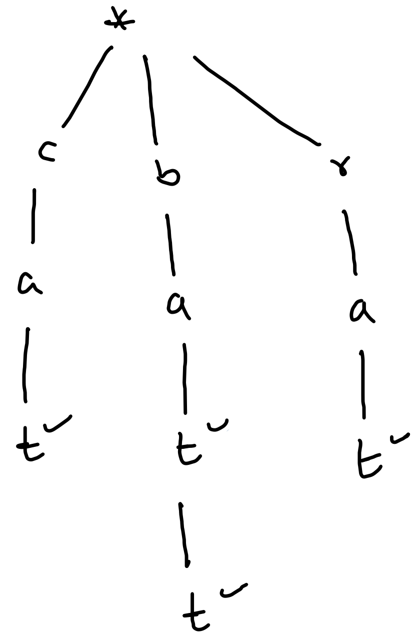
batt

```
Input: dictionary = ["cat","bat","rat"], sentence = "the cattle
was rattled by the battery"
Output: "the cat was rat by the bat"
```

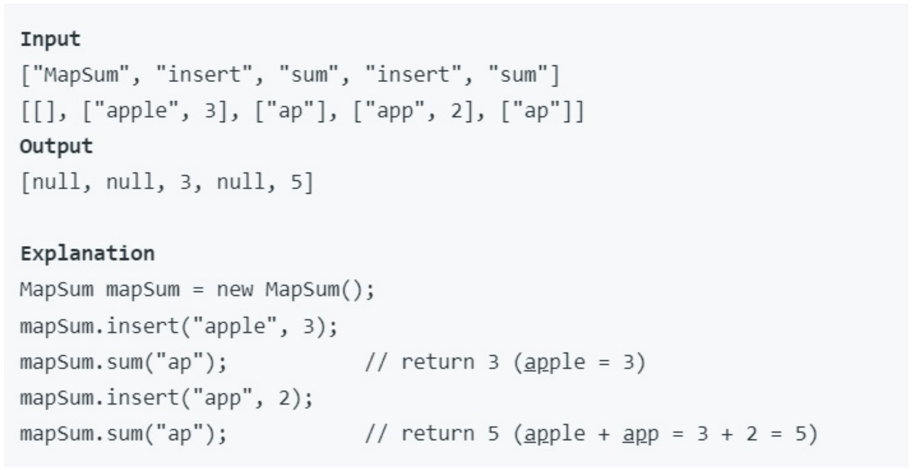
the $\overset{cat}{\cancel{cattle}}$ was $\overset{rat}{\cancel{rattled}}$ by the $\overset{bat}{\cancel{battery}}$

```
public String replaceWords(List<String> dictionary, String sent
    root = new Node();

    //fill all words of dictionary in "Trie"
    for(String word : dictionary) {
        insert(word);
    }

    String[]wordsArr = sentence.split(" ");

    for(int i=0; i < wordsArr.length;i++) {
        String prefix = searchPrefix(wordsArr[i],root);

        if(prefix != null) {
            wordsArr[i] = prefix;
        }
    }

    String ans = String.join(" ",wordsArr);
    return ans;
}
```

```
public String searchPrefix(String word,Node root) {
    Node curr = root;

    StringBuilder sb = new StringBuilder("");

    for(int i=0; i < word.length();i++) {
        char ch = word.charAt(i);

        if(curr.children[ch-'a'] == null) {
            return null;
        }

        sb.append(ch);
        curr = curr.children[ch-'a'];

        if(curr.isEnd == true) {
            return sb.toString();
        }

    }

    return null;
}
```



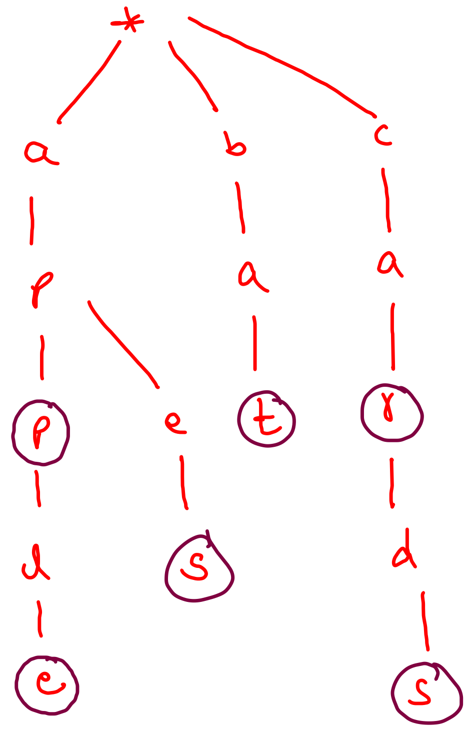
677. Map Sum Pairs

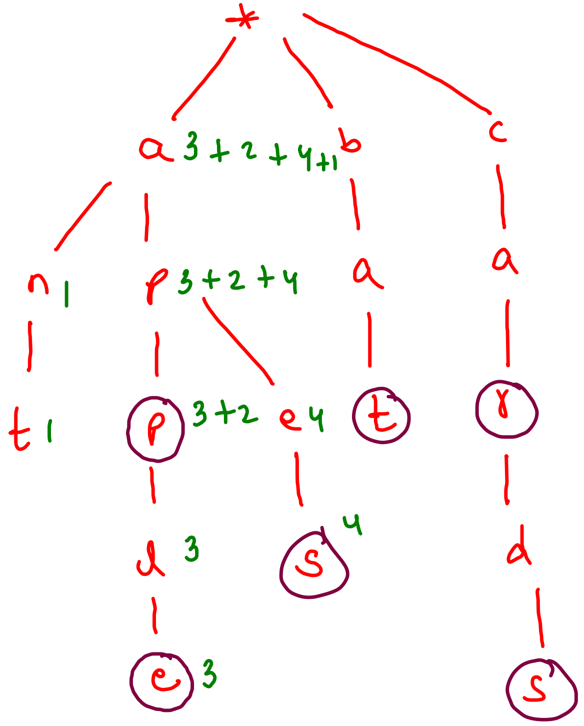


```
Input
["MapSum", "insert", "sum", "insert", "sum"]
[[], ["apple", 3], ["ap"], ["app", 2], ["ap"]]
Output
[null, null, 3, null, 5]

Explanation
MapSum mapSum = new MapSum();
mapSum.insert("apple", 3);
mapSum.sum("ap");           // return 3 (apple = 3)
mapSum.insert("app", 2);
mapSum.sum("ap");           // return 5 (apple + app = 3 + 2 = 5)
```

apple	3
app	2
apes	4
bat	3
cards	5
car	6

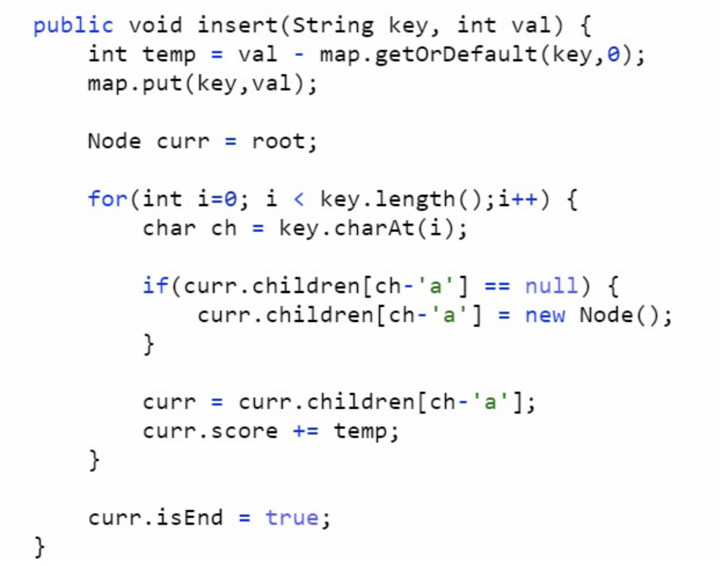
prefix = ap





ap.

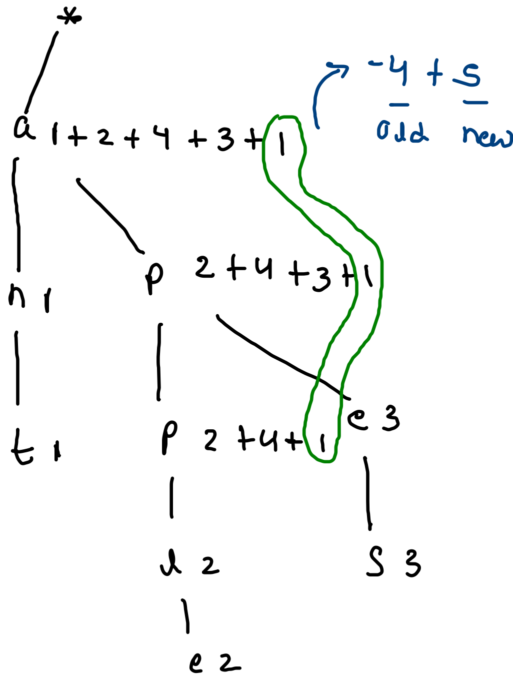
ant	1
apple	3
app	2
apes	4
bat	3
cards	5
car	6



```
public void insert(String key, int val) {
    int temp = val - map.getOrDefault(key,0);
    map.put(key,val);

    Node curr = root;

    for(int i=0; i < key.length();i++) {
        char ch = key.charAt(i);

        if(curr.children[ch-'a'] == null) {
            curr.children[ch-'a'] = new Node();
        }

        curr = curr.children[ch-'a'];
        curr.score += temp;
    }

    curr.isEnd = true;
}
```

ant - 1
apple - 2
app - 4
apes - 3
sum (ap) $\rightarrow$ 9
app - 5
sum (ap) $\rightarrow$ 10

ant = 1
apple = 2
app = 5
apes = 3



*
a 1+2+4+3+1
n 1
t 1
p 2+4+3+1
p 2+4+1
e 3
s 3
l 2
e 2
-4 + 5
old
new