# 1268. Search Suggestions System

## SOLUTION IN JAVA

class TrieNode {

public TrieNode[] children = new TrieNode[26];

public String word;

}

class Solution {

public List<List<String>> suggestedProducts(String[] products, String searchWord) {

List<List<String>> ans = new ArrayList<>();

for (final String product : products)

insert(product);

TrieNode node = root;

for (final char c : searchWord.toCharArray()) {

if (node == null || node.children[c - 'a'] == null) {

node = null;

ans.add(new ArrayList<>());

continue;

}

node = node.children[c - 'a'];

ans.add(search(node));

}

return ans;

}

private TrieNode root = new TrieNode();

private void insert(final String word) {

TrieNode node = root;

for (final char c : word.toCharArray()) {

final int i = c - 'a';

if (node.children[i] == null)

node.children[i] = new TrieNode();

node = node.children[i];

}

node.word = word;

}

private List<String> search(TrieNode node) {

List<String> res = new ArrayList<>();

dfs(node, res);

return res;

}

private void dfs(TrieNode node, List<String> ans) {

if (ans.size() == 3)

return;

if (node == null)

return;

if (node.word != null)

ans.add(node.word);

for (TrieNode child : node.children)

dfs(child, ans);

}

}