class TrieNode {

boolean isWord;

TrieNode[] children;

public TrieNode() {

isWord = false;

children = new TrieNode[26]; // 26 English lowercase letters

}

}

class Trie {

TrieNode root;

public Trie() {

root = new TrieNode();

}

public void insert(String word) {

TrieNode node = root;

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

int index = c - 'a';

if (node.children[index] == null) {

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

}

node = node.children[index];

}

node.isWord = true;

}

public boolean search(String word) {

TrieNode node = root;

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

int index = c - 'a';

if (node.children[index] == null) {

return false;

}

node = node.children[index];

}

return node.isWord;

}

public boolean startsWith(String prefix) {

TrieNode node = root;

for (char c : prefix.toCharArray()) {

int index = c - 'a';

if (node.children[index] == null) {

return false;

}

node = node.children[index];

}

return true;

}

}