TRIE TREE 模板

public class Trie {

TrieNode root;

/\*\* Initialize your data structure here. \*/

public Trie() {

root = new TrieNode();

}

/\*\* Inserts a word into the trie. \*/

public void insert(String word) {

char[] arr = word.toCharArray();

TrieNode curr = root;

for(int i = 0; i < arr.length; i++){

if(curr.map.get(arr[i]) == null){

curr.map.put(arr[i], new TrieNode(arr[i]));

}

curr = curr.map.get(arr[i]);

if(i == arr.length - 1){

curr.isWord = true;

}

}

}

/\*\* Returns if the word is in the trie. \*/

public boolean search(String word) {

TrieNode temp = startWith(word);

if(temp == null || ! temp.isWord){

return false;

}

return true;

}

/\*\* Returns if there is any word in the trie that starts with the given prefix. \*/

public boolean startsWith(String prefix) {

TrieNode temp = startWith(prefix);

if(temp == null)

return false;

else

return true;

}

private TrieNode startWith(String prefix){

TrieNode curr = root;

for(int i = 0; i < prefix.length(); i++){

char ch = prefix.charAt(i);

if(curr.map.get(ch) == null)

return null;

else{

curr = curr.map.get(ch);

}

}

return curr;

}

}

/\*\*

\* Your Trie object will be instantiated and called as such:

\* Trie obj = new Trie();

\* obj.insert(word);

\* boolean param\_2 = obj.search(word);

\* boolean param\_3 = obj.startsWith(prefix);

\*/

class TrieNode{

HashMap<Character, TrieNode> map;

char val;

boolean isWord;

public TrieNode(){

map = new HashMap<Character, TrieNode>();

isWord = false;

}

public TrieNode(char val){

map = new HashMap<Character, TrieNode>();

isWord = false;

val = val;

}

}