

Week 1: May 1st–May 7th

Problems appear at midnight, Pacific Time

Week 2: May 8th–May 14th

Problems appear at midnight, Pacific Time

- Check If It Is a Straight Line
- Valid Perfect Square
- Find the Town Judge
- Flood Fill
- Single Element in a Sorted ...
- Remove K Digits
- Implement Trie (Prefix Tree)

Week 3: May 15th–May 21st

The first problem for this section

Week 4: May 22nd–May 28th

## Implement Trie (Prefix Tree)

Solution

Implement a trie with `insert`, `search`, and `startsWith` methods.

Example:

```
Trie trie = new Trie();

trie.insert("apple");
trie.search("apple"); // returns true
trie.search("app"); // returns false
trie.startsWith("app"); // returns true
trie.insert("app");
trie.search("app"); // returns true
```

Note:

- You may assume that all inputs consist of lowercase letters `a-z`.
- All inputs are guaranteed to be non-empty strings.

Java



```
1 class TrieNode{
2     public Map<Character,TrieNode> map=new HashMap<>();
3     public boolean isEnd=false;
4     public void putChildIfAbsent(char ch){
5         map.putIfAbsent(ch,new TrieNode());
6     }
7     public TrieNode getChild(char ch){
8         return map.get(ch);
9     }
10 }
11 class Trie {
12     TrieNode root;
13     public Trie() {
14         root=new TrieNode();
15     }
16
17     /** Inserts a word into the trie. */
18     public void insert(String word) {
19         TrieNode curr=root;
20         for(char ch:word.toCharArray()){
21             curr.putChildIfAbsent(ch);
22             curr=curr.getChild(ch);
23         }
24         curr.isEnd=true;
25     }
26
27     /** Returns if the word is in the trie. */
28     public boolean search(String word) {
29         TrieNode curr=root;
30         for(char ch:word.toCharArray()){
31             curr=curr.getChild(ch);
32             if(curr==null) return false;
33         }
34         return curr.isEnd;
35     }
36
37     /** Returns if there is any word in the trie that starts with the given prefix. */
38     public boolean startsWith(String prefix) {
39         TrieNode curr=root;
40         for(char ch:prefix.toCharArray()){
41             curr=curr.getChild(ch);
42             if(curr==null) return false;
43         }
44         return true;
45     }
46 }
47
48 /**
49  * Your Trie object will be instantiated and called as such:
50  * Trie obj = new Trie();
51  * obj.insert(word);
52  * boolean param_2 = obj.search(word);
53  * boolean param_3 = obj.startsWith(prefix);
54  */
```

Custom Testcase (Contribute)



Run Code

Submit