75 Days of Code
Day 71
Problem no : Leetcode 1268
Problem Title : 1268. Search Suggestions System
Problem type : Trie

. Search Suggestions System

You are given an array of strings products and a string searchWord.

Design a system that suggests at most three product names from products after each character of searchWord is typed. Suggested products should have common prefix with searchWord. If there are more than three products with a common prefix return the three lexicographically minimums products.

Return a list of lists of the suggested products after each character of searchWord is typed.

Example 1:

Input: products = ["mobile","mouse","moneypot","monitor","mousepad"], searchWord = "mouse"
Output: [["mobile","moneypot","monitor"],["mobile","moneypot","monitor"],["mouse","mousepad"],["mouse","mousepad"],["mouse","mousepad"]]
Explanation: products sorted lexicographically = ["mobile","moneypot","monitor","mouse","mousepad"].
After typing m and mo all products match and we show user ["mobile","moneypot","monitor"].

After typing mou, mous and mouse the system suggests ["mouse","mousepad"].
Example 2:

Input: products = ["havana"], searchWord = "havana"
Output:
[["havana"],["havana"],["havana"],["havana"],["havana"],["havana"]]
Explanation: The only word "havana" will be always suggested while typing the search word.

```
You, 2 seconds ago | 1 author (You)
class TrieNode {
  children: Record<string, TrieNode>;
  products: string[] = [];
  constructor() {
    this.children = {};
    this.products = [];
  }
}

You, 2 seconds ago | 1 author (You)
class Trie {
  root: TrieNode;
  constructor() {       You, 1 second ago • Uncommitted changes
    this.root = new TrieNode();
  }
  insert(word: string): void {
    let node = this.root;
    for (const char of word) {
      if (!node.children[char]) {
        node.children[char] = new TrieNode();
      }
      node = node.children[char];
      node.products.push(word);
      node.products.sort();
      if (node.products.length > 3) {
        node.products.pop();
      }
    }
  }
  search(prefix: string): string[] {
    let node = this.root;
    for (const char of prefix) {
      if (!node.children[char]) {
        return [];
      }
      node = node.children[char];
    }
    return node.products;
  }
}
```

```typescript
function suggestedProducts(products: string[], searchWord: string): string[][] {
  let trie = new Trie();
  for (const product of products) {
    trie.insert(product);
  }
  const result: string[][] = [];
  let prefix = "";
  for (const char of searchWord) {
    prefix += char;
    const suggestion = trie.search(prefix);
    result.push([...suggestion]);
  }
  return result;
}
```

Accepted

Runtime	Details	Memory
220 ms		65.60 MB
Beats 18.18% of users with TypeScript		Beats 25.45% of users with TypeScript