

Search Suggestion System [LeetCode](#)

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.

Example: ["mobile", "mouse", "moneypot", "monitor", "mousepad"], searchedWord = "mouse"

Output: [["mobile", "moneypot", "monitor"], ["mobile", "moneypot", "monitor"], ["mouse", "mousepad"], ["mouse", "mousepad"], ["mouse", "mousepad"]]

Approach 1: Function to get suggested products for a list of products and a search word

- **TrieNode Class:**
 - Represents a node in the trie with children nodes, suggestions, and an end-of-word flag.
- **Trie Class:**
 - Manages the trie and provides methods to insert words and get suggestions.
 - Inserts each product into the trie, updating suggestions at each node.
- **getSuggestions Method:**
 - Retrieves suggestions for a given search word by traversing the trie.
 - Maintains a vector of suggestions at each step.
- **suggestedProducts Function:**
 - Sorts the list of products lexicographically.
 - Inserts the sorted products into the trie.
 - Calls the trie's getSuggestions method for the search word.

Time Complexity:

- **Insertion:** $O(N * L)$, where N is the number of products, and L is the average length of a product name.
- **Search (getSuggestions):** $O(L)$, where L is the length of the search word.

Space Complexity:

- **Trie Storage: $O(N * L)$, where N is the number of products, and L is the average length of a product name.**
- **Suggestions Vector: $O(L)$, where L is the length of the search word.**