

Started on Tuesday, 29 April 2025, 3:12 PM**State** Finished**Completed on** Tuesday, 29 April 2025, 3:22 PM**Time taken** 9 mins 21 secs**Marks** 4.00/10.00**Grade** 40.00 out of 100.00**Question 1**

Complete

Mark 0.00 out of 1.00

Which of the following statements about Tries is generally false?

- ☒ a. Tries offer worst-case search time that is better than a hash table.
- ☐ b. Tries can be used to implement auto-completion features.
- ☐ c. The height of a Trie is bounded by the length of the longest word stored.
- ☐ d. Tries are efficient for storing numeric data.

Question 2

Complete

Mark 0.00 out of 1.00

A Trie is used to store the words "ape", "apple", "apricot", and "bat". How many nodes in the Trie will have the character 'p' as part of their path from the root? (Include nodes where 'p' is the last character.)

- ☐ a. 3
- ☒ b. 4
- ☐ c. 2
- ☐ d. 1

Question 3

Complete

Mark 0.00 out of 1.00

In a Trie, which scenario leads to the most efficient search operation?

- ☐ a. Searching for a word that shares a long common prefix with many other words in the Trie.
- ☐ b. Searching for a word that has no common prefix with any other word in the Trie.
- ☐ c. Searching for a very long word that is not present in the Trie.
- ☒ d. Searching for a very short word that is present in the Trie.

Question 4

Complete

Mark 0.00 out of 1.00

What is the primary advantage of using a Trie over a standard hash table for storing a dictionary of words?

- ☐ a. More efficient storage of a small number of keys
- ☒ b. Ability to efficiently find all words with a given prefix
- ☐ c. Guaranteed constant-time search for all words
- ☐ d. Faster average-case insertion

Question 5

Complete

Mark 1.00 out of 1.00

Consider a Trie node. Which statement best describes the role of its 'children' attribute (a map of characters to Trie nodes)?

- ☐ a. It stores all possible next characters in the alphabet.
- ☒ b. It stores the next characters for all words that pass through this node.
- ☐ c. It stores the next characters for the longest word passing through this node
- ☐ d. It stores the next characters for all words that end at this node.

Question 6

Complete

Mark 0.00 out of 1.00

```
TrieNode root = new TrieNode();  
// Assume words "apple", "apricot", "banana" are inserted.  
String findPrefix(String word) {  
    TrieNode node = root;  
    StringBuilder prefix = new StringBuilder();  
    for (char c : word.toCharArray()) {  
        if (node.children.containsKey(c)) {  
            prefix.append(c);  
            node = node.children.get(c);  
        } else {  
            break;  
        }  
    }  
    return prefix.toString();  
}  
//What does findPrefix("apricot") return?
```

- ☒ a. "apricot"
- ☐ b. "a"
- ☐ c. "ap"
- ☐ d. ""

Question 7

Complete

Mark 0.00 out of 1.00

```
// Standard TrieNode structure
void insertWord(String word, TrieNode node, int index) {
    if (index == word.length()) {
        node.isEndOfWord = true;
        return;
    }
    char c = word.charAt(index);
    if (!node.children.containsKey(c)) {
        node.children.put(c, new TrieNode());
    }
    insertWord(word, node.children.get(c), index + 1);
}

// If insertWord("hello", root, 0) is called, how many recursive calls will be made to insertWord?
```

- ☐ a. 1
- ☒ b. 6
- ☐ c. 2
- ☐ d. 5

Question 8

Complete

Mark 1.00 out of 1.00

In a Trie used for autocomplete, how does the algorithm determine which words to suggest for a given prefix?

- ☐ a. It randomly selects words from the Trie.
- ☐ b. It uses a hash table to store words by prefix.
- ☒ c. It traverses the Trie path corresponding to the prefix and performs a depth-first search on the subtree rooted at the end of that path.
- ☐ d. It searches the entire Trie for words starting with the prefix.

Question 9

Complete

Mark 1.00 out of 1.00

```
boolean wordSearch(String word, TrieNode node, int index) {  
    if (index == word.length()) {  
        return node.isEndOfWord;  
    }  
    char c = word.charAt(index);  
    if (node.children.containsKey(c)) {  
        return wordSearch(word, node.children.get(c), index + 1);  
    }  
    return false;  
}
```

// What is the return value of wordSearch("car", root, 0) if only "car", "cart" and "cat" are inserted?

- ☐ a. null
- ☐ b. false
- ☒ c. true
- ☐ d. Depends on the root node

Question 10

Complete

Mark 1.00 out of 1.00

A Trie is used to store a large dictionary. Which optimization would most significantly reduce its memory footprint?

- ☒ a. Compressing non-branching paths into single edges (e.g., "appl" instead of 'a'-'>'p'-'>'p'-'>'l').
- ☐ b. Storing words in sorted order during insertion.
- ☐ c. Using a linked list instead of a HashMap for children.
- ☐ d. Using a smaller character set.