1. What is the time complexity for searching a key in a Trie with 'n' keys?
   1. O(n)
   2. O(log n)
   3. **O(k), where 'k' is the length of the key**
   4. O(1)
2. In a Trie, which data structure is commonly used to represent child nodes?
   1. Linked List
   2. **Array**
   3. Stack
   4. Queue
3. Which operation in a Trie is used to remove a key from the Trie?
   1. **delete()**
   2. remove()
   3. erase()
   4. pop()
4. What is the advantage of using a Trie over a Hash Table for string keys?
   1. Tries have faster insertion and deletion operations.
   2. **Tries have constant-time lookup for any key length.**
   3. Tries do not require a hash function.
   4. Tries use less memory.
5. In a compressed Trie (also called a Patricia Trie), how many children can a node have at most?
   1. 1
   2. 2
   3. **26 (for each alphabet letter)**
   4. Unlimited
6. Which operation can be used to find all keys in a Trie that have a common prefix?
   1. search()
   2. findPrefix()
   3. **startsWith()**
   4. retrievePrefix()
7. What is the primary disadvantage of using a Trie data structure?
   1. **High memory usage**
   2. Slow search operation
   3. Complexity in implementation
   4. Inefficient for small datasets
8. In a Trie, what is the purpose of a leaf node?
   1. To store the keys
   2. **To indicate the end of a valid key**
   3. To store metadata about the Trie
   4. To improve search performance
9. Which type of Trie optimizes space by merging nodes with a single child into a single node?
   1. **Compressed Trie**
   2. Radix Trie
   3. Suffix Trie
   4. Double Array Trie
10. What is the time complexity of inserting 'n' keys into an empty Trie?
    1. O(n)
    2. O(nlogn)
    3. O(nk), where 'k' is the average key length
    4. **O(k), where 'k' is the length of the longest key**
11. Which operation in a Trie is used to find the longest common prefix of a set of keys?
    1. **findLongestCommonPrefix()**
    2. search()
    3. startsWith()
    4. retrieveCommonPrefix()
12. In a Trie, what is the purpose of a null pointer in a child array?
    1. **It indicates the end of the Trie.**
    2. It represents a space character.
    3. It marks an invalid node.
    4. It is used for padding.
13. Which type of Trie is often used for implementing autocomplete functionality?
    1. **Radix Trie**
    2. Suffix Trie
    3. Compressed Trie
    4. Double Array Trie
14. In a Trie, what is the space complexity of storing 'n' keys?
    1. O(n)
    2. **O(nk), where 'k' is the average key length**
    3. O(1)
    4. O(k), where 'k' is the length of the longest key
15. Which operation in a Trie is used to count the number of keys with a given prefix?
    1. count()
    2. search()
    3. startsWith()
    4. **prefixCount()**
16. In a Trie, what is the purpose of the root node?
    1. To store the first character of all keys
    2. To improve search performance
    3. To indicate the end of a valid key
    4. **To serve as the entry point of the Trie**
17. What is the time complexity for deleting a key from a Trie with 'k' characters?
    1. **O(k)**
    2. O(log k)
    3. O(n)
    4. O(1)
18. Which data structure is typically used to implement a Trie node in practice?
    1. Array
    2. Linked List
    3. **Hashtable**
    4. Stack
19. In a Trie, what is the purpose of the path leading to a leaf node?
    1. **To represent a complete key**
    2. To store metadata about the Trie
    3. To indicate the end of a valid key
    4. To improve search performance
20. What is the primary advantage of using a Trie for pattern matching in strings?
    1. Fast insertion
    2. **Constant-time lookup**
    3. Efficient memory usage
    4. Easy implementation
21. In a Trie, what is the maximum number of children a node can have?
    1. 1
    2. 2
    3. **26 (for each alphabet letter)**
    4. Unlimited
22. Which operation is used to find the total number of keys stored in a Trie?
    1. **size()**
    2. count()
    3. length()
    4. keysCount()
23. What is the time complexity of searching for a key that does not exist in a Trie?
    1. O(1)
    2. **O(k), where 'k' is the length of the key**
    3. O(log n)
    4. O(n)
24. Which type of Trie is used for efficient pattern matching in a given text?
    1. Radix Trie
    2. **Suffix Trie**
    3. Compressed Trie
    4. Double Array Trie
25. What is the primary purpose of a Trie data structure?
    1. To store integers efficiently
    2. To implement a stack
    3. To perform sorting operations
    4. **To store and search for strings efficiently**
26. Which operation is used to check if a Trie is empty?
    1. **isEmpty()**
    2. hasKeys()
    3. isNull()
    4. checkEmpty()
27. Which type of Trie represents a compact form of a Trie, specifically designed for character-based operations?
    1. **Radix Trie**
    2. Compressed Trie
    3. Suffix Trie
    4. Patricia Trie
28. What is the primary advantage of using a Compressed Trie over a regular Trie?
    1. Faster insertion
    2. **Reduced memory usage**
    3. Constant-time lookup
    4. Improved search performance
29. Which operation in a Trie is used to find the number of keys with a common prefix?
    1. countPrefix()
    2. search()
    3. startsWith()
    4. **prefixKeysCount()**
30. Which operation is used to find the lexicographically smallest key in a Trie?
    1. **findSmallest()**
    2. search()
    3. startsWith()
    4. smallestKey()
31. Which type of Trie is used for efficiently storing a large set of keys in memory?
    1. Radix Trie
    2. Compressed Trie
    3. Suffix Trie
    4. **Patricia Trie**
32. What is the time complexity of inserting 'n' keys with an average key length of 'k' into a Trie?
    1. **O(nk)**
    2. O(n log n)
    3. O(k)
    4. O(n)
33. Which type of Trie is used for efficient storage and retrieval of keys in external memory (disk storage)?
    1. Radix Trie
    2. Suffix Trie
    3. **B-Trie**
    4. Double Array Trie
34. In a Trie, what is the advantage of using a hash map as the child node storage mechanism?
    1. Faster search operations
    2. Reduced memory usage
    3. Improved insertion speed
    4. **Support for non-alphabet characters**