100 Data Structure Interview QnA

Made by
Coding Hubs
Programmer

Want More
Search Coding Hubs
On Telegram & Join Channel

1. What is data structure?

A data structure is a way of organizing and storing data in a computer so that it can be accessed and manipulated efficiently.

2. What are the main types of data structures?

The main types of data structures are arrays, linked lists, stacks, queues, trees, graphs, and hash tables.

3. What is an array?

An array is a collection of elements of the same data type, stored in contiguous memory locations.

4. What is a linked list?

A linked list is a data structure where each element (node) contains a value and a pointer to the next element.

5. What is a stack?

A stack is a data structure that follows the Last-In-First-Out (LIFO) principle, where elements are added and removed from the top.

6. What is a queue?

A queue is a data structure that follows the First-In-First-Out (FIFO) principle, where elements are added at the rear and removed from the front.

7. What is a tree?

A tree is a hierarchical data structure composed of nodes, where each node can have zero or more child nodes.

8. What is a binary tree?

A binary tree is a tree in which each node has at most two child nodes, referred to as the left child and the right child.

9. What is a graph?

A graph is a collection of nodes (vertices) and edges, where edges represent relationships between nodes.

8. What is a binary tree?

A binary tree is a tree in which each node has at most two child nodes, referred to as the left child and the right child.

9. What is a graph?

A graph is a collection of nodes (vertices) and edges, where edges represent relationships between nodes.

10 What is a hash table?

A hash table is a data structure that uses a hash function to map keys to values, allowing for efficient retrieval and insertion.

11. What is the difference between an array and a linked list?

An array has a fixed size, requires contiguous memory, and allows for random access, whereas a linked list can dynamically grow and doesn't require contiguous memory but doesn't support random access.

12. What is the time complexity of accessing an element in an array?

The time complexity of accessing an element in an array is O(1) (constant time).

13. What is the time complexity of searching an element in a linked list?

The time complexity of searching an element in a linked list is O(n) (linear time).

14. What is a doubly linked list?

A doubly linked list is a linked list in which each node contains two pointers, one pointing to the previous node and one pointing to the next node.

15. What is a circular linked list?

A circular linked list is a linked list where the last node points back to the first node, forming a loop.

16. What operations can be performed on a stack?

The main operations on a stack are push (to add an element), pop (to remove the top element), and peek (to view the top element without removing it).

17. What operations can be performed on a queue?

The main operations on a queue are enqueue (to add an element), dequeue (to remove the front element), and peek (to view the front element without removing it).

18. What is a binary search tree?

A binary search tree is a binary tree in which the left child of a node contains a value less than or equal to the node's value, and the right child contains a value greater than the node's value.

19. What is an AVL tree?

An AVL tree is a self-balancing binary search tree in which the heights of the left and right subtrees differ by at most one.

20. What is a hash function?

A hash function is a function that takes an input (key) and produces a fixed-size value (hash code), which is used to index the data in a hash table.

21. What is collision handling in a hash table?

Collision handling is the process of dealing with situations where two different keys map to the same hash code. Common techniques include chaining (using linked lists) and open addressing.

22. What is the time complexity of searching in a balanced binary search tree?

The time complexity of searching in a balanced binary search tree, such as an AVL tree, is O(log n) (logarithmic time).

23. What is a breadth-first search (BFS)?

Breadth-first search is a graph traversal algorithm that explores all the vertices of a graph in breadth-first order, visiting all the neighbors of a node before moving to the next level.

24. What is a depth-first search (DFS)?

Depth-first search is a graph traversal algorithm that explores as far as possible along each branch before backtracking, visiting all the descendants of a node before moving to the next sibling.

25. What is a heap?

A heap is a complete binary tree that satisfies the heap property, which means that each parent node is either greater than or equal to (in a max heap) or less than or equal to (in a min heap) its children.

26. What is the difference between a binary tree and a binary search tree?

A binary tree is a tree in which each node can have at most two child nodes, whereas a binary search tree is a binary tree that satisfies the property that the left child of a node is less than or equal to the node, and the right child is greater than or equal to the node.

27. What is the time complexity of inserting an element into a heap?

The time complexity of inserting an element into a heap is O(log n) (logarithmic time).

28. What is the time complexity of removing the root element from a heap?

The time complexity of removing the root element from a heap is O(log n) (logarithmic time).

29. What is a trie?

A trie, also known as a prefix tree, is a tree-like data structure that stores a collection of strings, where each node represents a common prefix.

30. What is the time complexity of searching in a trie?

The time complexity of searching in a trie is O(m), where m is the length of the search key.

31. What is dynamic programming?

Dynamic programming is a technique for solving complex problems by breaking them down into simpler overlapping subproblems and solving them only once, storing the results to avoid redundant computations.

32. What is the difference between an array and a matrix?

An array is a one-dimensional data structure, whereas a matrix is a two-dimensional data structure with rows and columns.

33. What is a sparse matrix?

A sparse matrix is a matrix in which most of the elements are zero.

34. What is a self-balancing binary search tree?

A self-balancing binary search tree is a binary search tree that automatically adjusts its structure to maintain a balance, ensuring efficient search, insertion, and deletion operations.

35. What is a B-tree?

A B-tree is a self-balancing search tree that can have multiple keys and children per node, designed to reduce the number of disk accesses required for large datasets.

36. What is the time complexity of searching in a B-tree?

The time complexity of searching in a B-tree is O(log n) (logarithmic time).

37. What is a red-black tree?

A red-black tree is a self-balancing binary search tree in which each node is colored red or black, and a set of properties are maintained to ensure balance.

38. What is the time complexity of searching in a red-black tree?

The time complexity of searching in a red-black tree is O(log n) (logarithmic time).

39. What is the difference between a stack and a queue?

A stack follows the Last-In-First-Out (LIFO) principle, while a queue follows the First-In-First-Out (FIFO) principle.

40. What is the time complexity of searching in a hash table?

The average time complexity of searching in a hash table is O(1) (constant time), but it can be O(n) in the worst case if there are many collisions.

41. What is the difference between a singly linked list and a doubly linked list?

In a singly linked list, each node contains a pointer to the next node, while in a doubly linked list, each node contains pointers to both the next and previous nodes.

42. What is a circular queue?

A circular queue is a queue implemented in a circular manner, where the last element points back to the first element, allowing for efficient memory utilization.

43. What is a priority queue?

A priority queue is an abstract data type that supports inserting elements with associated priorities and retrieving the element with the highest (or lowest) priority.

44. What is the time complexity of inserting an element into a priority queue?

The time complexity of inserting an element into a priority queue depends on the implementation. In a binary heap-based priority queue, the time complexity is O(log n) (logarithmic time).

45. What is a self-balancing binary search tree?

A self-balancing binary search tree is a binary search tree that automatically adjusts its structure to maintain balance, ensuring efficient search, insertion, and deletion operations.

46. What is the difference between a stack and a heap?

A stack is a region of memory used for automatic storage of variables and function call information, while a heap is a region of memory used for dynamic memory allocation.

47. What is the time complexity of searching in a self-balancing binary search tree?

The time complexity of searching in a self-balancing binary search tree, such as an AVL tree or red-black tree, is O(log n) (logarithmic time).

48. What is a circular linked list?

A circular linked list is a linked list in which the last node points back to the first node, forming a loop.

49. What is the time complexity of inserting an element into a linked list?

The time complexity of inserting an element into a linked list at a specific position is O(1) (constant time) if the position is known, or O(n) (linear time) if the position needs to be searched.

50. What is a skip list?

A skip list is a data structure that allows for efficient search, insertion, and deletion operations, resembling a linked list with multiple parallel layers.

51. What is the time complexity of searching in a skip list?

The time complexity of searching in a skip list is O(log n) (logarithmic time).

52. What is a disjoint set data structure?

A disjoint set data structure is a data structure that keeps track of a collection of disjoint (non-overlapping) sets and supports efficient union and find operations.

53. What is the time complexity of finding the root of a disjoint set?

The time complexity of finding the root of a disjoint set using the find operation is typically O(α(n)), where α is the inverse Ackermann function (a very slowly growing function).

54. What is a suffix tree?

A suffix tree is a data structure that represents all the suffixes of a given string in a compressed form, enabling efficient pattern matching and substring search operations.

55. What is the time complexity of building a suffix tree?

The time complexity of building a suffix tree for a string of length n is O(n).

56. What is a bit array?

A bit array, also known as a bit set or bit vector, is a data structure that represents an array of bits, where each bit can be set (1) or cleared (0), allowing for efficient storage and manipulation of binary data.

57. What is the time complexity of inserting an element into a bit array?

The time complexity of inserting an element into a bit array depends on the specific operation being performed.

58. What is a bloom filter?

A bloom filter is a probabilistic data structure that tests whether an element is a member of a set, with a small possibility of false positives but no false negatives.

59. What is the time complexity of inserting an element into a bloom filter?

The time complexity of inserting an element into a bloom filter is O(k), where k is the number of hash functions used.

60. What is a Fibonacci heap?

A Fibonacci heap is a data structure that supports efficient insertion, deletion, and extraction of the minimum element, making it suitable for certain graph algorithms.

61. What is the time complexity of extracting the minimum element from a Fibonacci heap?

The time complexity of extracting the minimum element from a Fibonacci heap is O(log n) amortized time, where n is the number of elements in the heap.

62. What is a disjoint-set forest?

A disjoint-set forest is a data structure that represents a collection of disjoint sets, where each set is represented by a rooted tree.

63. What is the time complexity of finding the root of a disjoint-set forest?

The time complexity of finding the root of a disjoint-set forest using the find operation with path compression is nearly O(1) (constant time).

64. What is a van Emde Boas tree?

A van Emde Boas tree is a tree-like data structure that provides efficient operations for a universe of size M, achieving a time complexity of O(log log M).

65. What is the time complexity of searching in a van Emde Boas tree?

The time complexity of searching in a van Emde Boas tree is O(log log M), where M is the size of the universe.

66. What is a Fenwick tree (Binary Indexed Tree)?

A Fenwick tree, also known as a Binary Indexed Tree (BIT), is a data structure that allows efficient prefix sum calculations and single element updates in an array.

67. What is the time complexity of calculating a prefix sum using a Fenwick tree?

The time complexity of calculating a prefix sum using a Fenwick tree is O(log n), where n is the size of the array.

68. What is a range tree?

A range tree is a tree-like data structure that allows efficient range queries, such as finding all elements within a given range.

69. What is the time complexity of searching in a range tree?

The time complexity of searching in a range tree is O(log n + k), where n is the number of elements and k is the number of elements within the range.

70. What is a segment tree?

A segment tree is a tree-like data structure that allows efficient range queries and updates on an array.

71. What is the time complexity of searching in a segment tree?

The time complexity of searching in a segment tree is O(log n + k), where n is the size of the array and k is the number of elements within the range.

72. What is a trie (prefix tree)?

A trie, also known as a prefix tree, is a tree-like data structure that stores a collection of strings, where each node represents a common prefix.

73. What is the time complexity of searching in a trie?

The time complexity of searching in a trie is O(m), where m is the length of the search key.

74. What is a kd-tree?

A kd-tree is a binary tree that is used to organize points in k-dimensional space, allowing efficient nearest neighbor search operations.

75. What is the time complexity of searching in a kd-tree?

The time complexity of searching in a balanced kd-tree is typically O(log n), where n is the number of points.

76. What is a bloom filter?

A bloom filter is a probabilistic data structure that tests whether an element is a member of a set, with a small possibility of false positives but no false negatives.

77. What is the time complexity of inserting an element into a bloom filter?

The time complexity of inserting an element into a bloom filter is O(k), where k is the number of hash functions used.

78. What is a trie (prefix tree)?

A trie, also known as a prefix tree, is a tree-like data structure that stores a collection of strings, where each node represents a common prefix.

79. What is the time complexity of searching in a trie?

The time complexity of searching in a trie is O(m), where m is the length of the search key.

80. What is an interval tree?

An interval tree is a tree-like data structure used to efficiently store and search for intervals, enabling range-based queries.

81. What is the time complexity of searching in an interval tree?

The time complexity of searching in an interval tree is typically O(log n + k), where n is the number of intervals and k is the number of intervals intersecting the search range.

82. What is a radix tree?

A radix tree, also known as a Patricia trie or compact prefix tree, is a compressed trie data structure used for efficient storage and retrieval of strings with common prefixes.

83. What is the time complexity of searching in a radix tree?

The time complexity of searching in a radix tree is O(m), where m is the length of the search key.

84. What is a heap?

A heap is a complete binary tree that satisfies the heap property, which means that each parent node is either greater than or equal to (in a max heap) or less than or equal to (in a min heap) its children.

85. What is the time complexity of inserting an element into a heap?

The time complexity of inserting an element into a heap is O(log n) (logarithmic time).

86. What is a monotonic stack?

A monotonic stack is a stack data structure where the elements are either in non-increasing or non-decreasing order, allowing for efficient computations on certain types of sequences.

87. What is the time complexity of finding the next greater element using a monotonic stack?

The time complexity of finding the next greater element using a monotonic stack is O(n), where n is the number of elements in the sequence.

88. What is a trie (prefix tree)?

A trie, also known as a prefix tree, is a tree-like data structure that stores a collection of strings, where each node represents a common prefix.

89. What is the time complexity of searching in a trie?

The time complexity of searching in a trie is O(m), where m is the length of the search key.

90. What is a skip list?

A skip list is a data structure that allows for efficient search, insertion, and deletion operations, resembling a linked list with multiple parallel layers.

91. What is the time complexity of searching in a skip list?

The time complexity of searching in a skip list is O(log n) (logarithmic time).

92. What is a sparse matrix?

A sparse matrix is a matrix in which most of the elements are zero.

93. What is a self-balancing binary search tree?

A self-balancing binary search tree is a binary search tree that automatically adjusts its structure to maintain balance, ensuring efficient search, insertion, and deletion operations.

94. What is the difference between a stack and a heap?

A stack is a region of memory used for automatic storage of variables and function call information, while a heap is a region of memory used for dynamic memory allocation.

95. What is the time complexity of searching in a self-balancing binary search tree?

The time complexity of searching in a self-balancing binary search tree, such as an AVL tree or red-black tree, is O(log n) (logarithmic time).

96. What is a circular linked list?

A circular linked list is a linked list in which the last node points back to the first node, forming a loop.

97. What is a priority queue?

A priority queue is an abstract data type that supports inserting elements with associated priorities and retrieving the element with the highest (or lowest) priority.

98. What is the time complexity of inserting an element into a priority queue?

The time complexity of inserting an element into a priority queue depends on the implementation. In a binary heap-based priority queue, the time complexity is O(log n) (logarithmic time).

99. What is a disjoint set data structure?

A disjoint set data structure is a data structure that keeps track of a collection of disjoint (non-overlapping) sets and supports efficient union and find operations.

100. What is the time complexity of finding the root of a disjoint set?

The time complexity of finding the root of a disjoint set using the find operation is typically $O(\alpha(n))$, where α is the inverse Ackermann function (a very slowly growing function).