Practice MCQS

1.What is a binary tree?
A. A tree with exactly two children for each node
B. A tree with at most two children for each node
C. A tree with one child for each node
D. A tree with no child for each node

Answer: B. A tree with at most two children for each node

2. Which of the following statements about binary trees is true?
A. All binary trees are binary search trees
B. All binary search trees are binary trees
C. Binary trees and binary search trees are the same
D. Binary trees have a height equal to the number of nodes in the tree

Answer: B. All binary search trees are binary trees

3.What is the maximum number of nodes in level 3 in a binary tree?
A. 3
B. 4
C. 7
D. 8

Answer: C. 7

4. In a binary tree, a node with no children is called:
A. Parent node
B. Leaf node
C. Root node
D. Sibling node

Answer: B. Leaf node

5.Which traversal visits the root node after traversing the left subtree and before traversing the right subtree?

A. Preorder
B. Inorder

C. Postorder
D. Level order

Answer: B. Inorder

6. In an inorder traversal of a binary search tree, the nodes are visited in:
A. Non-decreasing order
B. Non-increasing order
C. Random order
D. Order of insertion

Answer: A. Non-decreasing order

7. Which of the following operations cannot be performed in O(log n) time in a binary search tree?
A. Search
B. Insertion
C. Deletion
D. Traversal

Answer: D. Traversal

8. Which of the following traversal algorithms uses a stack data structure?
A. Inorder
B. Preorder
C. Postorder
D. Level order

Answer: A. Inorder

9. Which of the following trees is not a binary tree?
A. Complete Binary Tree
B. Full Binary Tree
C. Perfect Binary Tree
D. Trie

Answer: D. Trie

10. The maximum number of nodes in a binary tree of height h is:

A. 2^h - 1
B. h^2
C. 2 * h
D. h!

Answer: A. 2^h - 1

11. What is a binary search tree (BST)?
A. A tree with exactly two children for each node
B. A tree with at most two children for each node
C. A binary tree where each node's value is greater than its left subtree and less than its right subtree
D. A binary tree where each node's value is less than its left subtree and greater than its right subtree

Answer: C. A binary tree where each node's value is greater than its left subtree and less than its right subtree

12. In a binary search tree, which traversal visits the nodes in sorted order?
A. Preorder
B. Inorder
C. Postorder
D. Level order

Answer: B. Inorder

13. What is the time complexity of searching for a value in a balanced binary search tree?
A. O(1)
B. O(log n)
C. O(n)
D. O(n log n)

Answer: B. O(log n)

14. Which of the following operations in a binary search tree may require tree re-balancing to maintain its properties?
A. Search
B. Insertion

C. Deletion
D. Traversal

Answer: C. Deletion

15. The minimum value in a binary search tree is found:
A. At the root node
B. At the leftmost leaf node
C. At the rightmost leaf node
D. At any random node

Answer: B. At the leftmost leaf node

17. In a binary search tree, which operation has the worst-case time complexity?
A. Search
B. Insertion
C. Deletion
D. Traversal

Answer: D. Traversal

18. Which of the following is not a valid property of a binary search tree?
A. All nodes in the left subtree are less than the root node.
B. All nodes in the right subtree are greater than the root node.
C. All nodes in the left subtree are less than or equal to the root node.
D. All nodes in the right subtree are greater than or equal to the root node.

Answer: C. All nodes in the left subtree are less than or equal to the root node.

19. Which traversal of a binary search tree returns the nodes in descending order?
A. Preorder
B. Inorder
C. Postorder
D. Reverse Inorder

Answer: D. Reverse Inorder

21. What is a heap priority queue?
A. A data structure that maintains elements in sorted order
B. A binary tree where each node's value is greater than its children
C. A data structure that maintains the maximum (or minimum) element at the root
D. A data structure that maintains elements in a random order

Answer: C. A data structure that maintains the maximum (or minimum) element at the root

22. Which operation has a time complexity of O(log n) in a binary heap?
A. Insertion
B. Deletion
C. Searching
D. Traversal

Answer: A. Insertion

23. In a max-heap, which element will be at the root?
A. The maximum element
B. The minimum element
C. Any random element
D. The median element

Answer: A. The maximum element

24. Which of the following data structures is typically used to implement a heap priority queue?
A. Array
B. Linked List
C. Stack
D. Queue

Answer: A. Array

25. What is the height of a heap containing n elements?
A. log(n)
B. n
C. log(n) + 1
D. n/2

Answer: A. log(n)

27. In a min-heap, which element will be at the root?
A. The maximum element
B. The minimum element
C. Any random element
D. The median element

Answer: B. The minimum element

28. Which operation is used to insert an element into a heap priority queue?
A. push()
B. pop()
C. enqueue()
D. dequeue()

Answer: A. push()

29. Which of the following heap operations requires restructuring the heap to maintain its properties?
A. Insertion
B. Deletion
C. Searching
D. Traversal

Answer: B. Deletion

30. What is the time complexity of building a heap from an array of n elements?
A. O(n)
B. O(log n)
C. O(n log n)
D. O(n^2)

Answer: C. O(n log n)

31. What is a graph?

A. A data structure representing hierarchical relationships
B. A data structure representing a collection of key-value pairs
C. A data structure representing pairwise relationships between objects
D. A data structure representing a linear sequence of elements

Answer: C. A data structure representing pairwise relationships between objects

32. Which of the following is not a type of graph?
A. Directed graph
B. Undirected graph
C. Bipartite graph
D. Sequential graph

Answer: D. Sequential graph

33. What is the degree of a vertex in a graph?
A. The number of vertices connected to it
B. The number of edges connected to it
C. The distance from the root vertex
D. The value assigned to the vertex

Answer: B. The number of edges connected to it

34. Which of the following statements is true about a directed graph?
A. Every edge is bidirectional
B. There is no concept of direction in a directed graph
C. Edges have a direction from one vertex to another
D. All vertices have the same degree

Answer: C. Edges have a direction from one vertex to another

35. In an undirected graph, the number of edges is typically expressed in terms of:
A. The number of vertices
B. The number of components
C. The degree of the vertices
D. The number of connected pairs of vertices

Answer: A. The number of vertices

36. What is a cycle in a graph?

A. A path that starts and ends at the same vertex
B. A path that visits every vertex exactly once
C. A path with a length greater than a certain threshold
D. A path that contains a specific sequence of vertices

Answer: A. A path that starts and ends at the same vertex

37. Which of the following algorithms is used to find the shortest path between two vertices in a weighted graph?
A. Depth First Search (DFS)
B. Breadth First Search (BFS)
C. Dijkstra's algorithm
D. Kruskal's algorithm

Answer: C. Dijkstra's algorithm

38. What is the time complexity of the depth-first search (DFS) algorithm for traversing a graph with n vertices and m edges?
A. O(n)
B. O(m)
C. O(n + m)
D. O(n * m)

Answer: C. O(n + m)

39. Which of the following graph representations is most suitable for sparse graphs?
A. Adjacency Matrix
B. Adjacency List
C. Incidence Matrix
D. Edge List

Answer: B. Adjacency List

40. In a weighted graph, what does the weight of an edge represent?
A. The number of vertices connected by the edge
B. The distance between the vertices connected by the edge
C. The color assigned to the edge
D. The degree of the vertices connected by the edge

Answer: B. The distance between the vertices connected by the edge

41. What is a spanning tree of a graph?
A. A tree that contains all the vertices of the graph
B. A tree with the minimum number of edges that connects all vertices of the graph
C. A tree with the maximum number of edges that connects all vertices of the graph
D. A tree with a specific height in the graph

Answer: B. A tree with the minimum number of edges that connects all vertices of the graph

42. Which algorithm is used to find the minimum spanning tree in a weighted graph?
A. Depth First Search (DFS)
B. Breadth First Search (BFS)
C. Kruskal's algorithm
D. Dijkstra's algorithm

Answer: C. Kruskal's algorithm

43. Which of the following is not a property of a tree?
A. Acyclic
B. Connected
C. Directed
D. Single root

Answer: C. Directed

45. Which graph algorithm is used to detect cycles in a graph?
A. Depth First Search (DFS)
B. Breadth First Search (BFS)
C. Dijkstra's algorithm
D. Floyd-Warshall algorithm

Answer: A. Depth First Search (DFS)

46. Which of the following is not a traversal algorithm for graphs?
A. Depth First Search (DFS)
B. Breadth First Search (BFS)
C. Preorder Traversal
D. Inorder Traversal

Answer: D. Inorder Traversal

47. Which of the following data structures is used to implement Breadth First Search (BFS)?
A. Queue
B. Stack
C. Array
D. Linked List

Answer: A. Queue

48. In a directed graph, a path from vertex A to vertex B exists if:
A. There is an edge from A to B
B. There is an edge from B to A
C. There is a sequence of vertices starting from A and ending at B
D. There is a sequence of vertices starting from B and ending at A

Answer: C. There is a sequence of vertices starting from A and ending at B

49. What is the minimum number of edges required to form a cycle in a connected undirected graph with n vertices?
A. n - 1
B. n
C. n + 1
D. 2

Answer: B. n

50. Which of the following graph representations is most suitable for dense graphs?
A. Adjacency Matrix
B. Adjacency List
C. Incidence Matrix
D. Edge List

Answer: A. Adjacency Matrix

51. What is a HashMap?
A. A data structure that stores elements in a sorted order
B. A data structure that stores elements in an unsorted order
C. A data structure that stores key-value pairs with unique keys
D. A data structure that stores key-value pairs with duplicate keys

Answer: C. A data structure that stores key-value pairs with unique keys

52. Which of the following operations in a HashMap has constant time complexity on average?
A. Insertion
B. Deletion
C. Search
D. Traversal

Answer: C. Search

53. In a HashMap, the key is used to:
A. Determine the size of the HashMap
B. Determine the position where the value is stored
C. Determine the hash code of the value
D. Determine the data type of the value

Answer: C. Determine the hash code of the value

54. What happens if you try to insert a duplicate key into a HashMap?
A. The value associated with the existing key is updated
B. An error is thrown
C. The new key-value pair is added as a separate entry
D. The operation fails and returns false

Answer: A. The value associated with the existing key is updated

55. Which method is used to retrieve the value associated with a key in a HashMap?
A. get()
B. put()
C. remove()
D. containsKey()

Answer: A. get()

56. What is the time complexity of the get() operation in a HashMap?
A. O(1)
B. O(log n)
C. O(n)

D. O(n log n)

Answer: A. O(1)

57. Which of the following implementations of HashMap is not synchronized?
A. HashMap
B. LinkedHashMap
C. TreeMap
D. ConcurrentHashMap

Answer: A. HashMap

58. What is the purpose of the hash function in a HashMap?
A. To generate unique keys for each value
B. To convert the key into an index for storage
C. To encrypt the key-value pairs
D. To compress the size of the HashMap

Answer: B. To convert the key into an index for storage

59. What is the default initial capacity of a HashMap?
A. 10
B. 16
C. 32
D. 64

Answer: B. 16

60. Which data structure is internally used to implement HashMap in most programming languages?
A. Array
B. Linked List
C. Binary Tree
D. Hash Table

Answer: A. Array

61. How does a HashMap handle collisions?
A. By replacing the existing value with the new value

B. By chaining (using linked lists or other data structures) to store multiple values at the same index
C. By resizing the HashMap to accommodate more elements
D. By ignoring the new value if a collision occurs

Answer: B. By chaining (using linked lists or other data structures) to store multiple values at the same index

62. What is the load factor of a HashMap?
A. The ratio of the number of elements to the total capacity of the HashMap
B. The ratio of the total capacity of the HashMap to the number of elements
C. The number of elements stored in the HashMap
D. The total capacity of the HashMap

Answer: A. The ratio of the number of elements to the total capacity of the HashMap

63. Which of the following operations in a HashMap has a time complexity of O(1) on average?
A. Insertion
B. Deletion
C. Search
D. Traversal

Answer: A. Insertion

64. What happens if you try to retrieve a value associated with a non-existent key in a HashMap using the get() method?
A. An error is thrown
B. The method returns null
C. The method returns -1
D. The method returns an empty string

Answer: B. The method returns null

65. In a HashMap, what happens if the load factor exceeds a certain threshold?
A. The HashMap is resized to increase its capacity
B. The HashMap is converted into a TreeMap
C. The HashMap is converted into a LinkedHashMap
D. The HashMap becomes read-only

Answer: A. The HashMap is resized to increase its capacity

66. What is a Trie?
A. A tree data structure used to store elements in sorted order
B. A tree data structure used to store key-value pairs
C. A tree data structure used to store strings efficiently
D. A tree data structure used to store integers efficiently

Answer: C. A tree data structure used to store strings efficiently

67. What is the time complexity for searching a string in a Trie?
A. O(1)
B. O(log n)
C. O(m), where m is the length of the string
D. O(n), where n is the number of strings stored in the Trie

Answer: C. O(m), where m is the length of the string

68. Which operation is efficient in a Trie for determining if a string is a prefix of any other string?
A. Insertion
B. Deletion
C. Searching
D. Traversal

Answer: C. Searching

69. What is the structure of a Trie node?
A. Each node has a single character and a pointer to its parent node
B. Each node has a single character and an array of pointers to its child nodes
C. Each node has multiple characters and a pointer to its parent node
D. Each node has multiple characters and an array of pointers to its child nodes

Answer: D. Each node has multiple characters and an array of pointers to its child nodes

70. In a Trie, how many children can a node have?
A. At most one child
B. At most two children

C. At most 26 children (for lowercase English alphabets)
D. Unlimited number of children

Answer: C. At most 26 children (for lowercase English alphabets)

71. Which of the following operations in a Trie has a time complexity of O(m), where m is the length of the string being inserted?
A. Searching
B. Insertion
C. Deletion
D. Traversal
Answer: B. Insertion

72. What is the space complexity of a Trie?
A. O(1)
B. O(log n)
C. O(m), where m is the length of the longest string
D. O(n), where n is the number of strings stored in the Trie

Answer: C. O(m), where m is the length of the longest string

73. Which of the following is a disadvantage of using a Trie?
A. It requires less memory compared to other data structures
B. It has slower search operations compared to other data structures
C. It is more complex to implement compared to other data structures
D. It cannot efficiently handle dynamic sets of strings

Answer: C. It is more complex to implement compared to other data structures

74. What is the time complexity of deleting a string from a Trie?
A. O(1)
B. O(log n)
C. O(m), where m is the length of the string
D. O(n), where n is the number of strings stored in the Trie

Answer: C. O(m), where m is the length of the string

75. Which data structure is typically used to implement the child nodes in a Trie?
A. Array

B. Linked List
C. Binary Tree
D. Hash Map

Answer: A. Array

76. What is a stack?

A. A data structure that follows FIFO (First-In-First-Out) order
B. A data structure that follows LIFO (Last-In-First-Out) order
C. A data structure that allows random access to elements
D. A data structure that sorts elements automatically

Answer: B. A data structure that follows LIFO (Last-In-First-Out) order

77. Which operation adds an element to the top of the stack?

A. push()
B. pop()
C. top()
D. dequeue()

Answer: A. push()

78. What is the time complexity of the push() operation in a stack?

A. O(1)
B. O(log n)
C. O(n)
D. O(n log n)

Answer: A. O(1)

79. Which operation removes the top element from the stack?

A. push()
B. pop()
C. peek()
D. dequeue()

Answer: B. pop()

80. In a stack, which element is accessed first when performing the pop() operation?
A. The top element
B. The bottom element
C. The middle element
D. Any random element

Answer: A. The top element

81. What is a queue?

A. A data structure that follows FIFO (First-In-First-Out) order
B. A data structure that follows LIFO (Last-In-First-Out) order
C. A data structure that allows random access to elements
D. A data structure that sorts elements automatically

Answer: A. A data structure that follows FIFO (First-In-First-Out) order

82. Which operation adds an element to the rear of the queue?

A. enqueue()
B. dequeue()
C. push()
D. pop()

Answer: A. enqueue()

83. What is the time complexity of the enqueue() operation in a queue?
A. O(1)
B. O(log n)
C. O(n)
D. O(n log n)

Answer: A. O(1)

84. Which operation removes an element from the front of the queue?
A. enqueue()

B. dequeue()
C. push()
D. pop()

Answer: B. dequeue()

85. In a queue, which element is accessed first when performing the dequeue() operation?

A. The front element
B. The rear element
C. The middle element
D. Any random element

Answer: A. The front element

86. How do you access the fifth element in an array named arr?

a) arr[4]
b) arr[5]
c) arr[0]
d) arr[1]

Answer: a) arr[4]

87. In C++, how can you find the number of elements in an array named numbers?

a) numbers.length()
b) sizeof(numbers)
c) numbers.size()
d) sizeof(numbers) / sizeof(numbers[0])

Answer: d) sizeof(numbers) / sizeof(numbers[0])

88. What is the purpose of the memset function in C++?

a) To calculate the size of an array
b) To set all elements of an array to a specific value

c) To reverse the elements of an array
d) To find the maximum element in an array

Answer: b) To set all elements of an array to a specific value

89. In C++, what is the correct way to declare a string?

a) string name = "John";
b) char name[] = "John";
c) char name[5] = "John";
d) String name = "John";

Answer: a) string name = "John";

90. How do you find the length of a C-style string (char array) in C++?

a) Using strlen() function
b) Using length() function
c) Using size() function
d) Using sizeof() operator

Answer: a) Using strlen() function

91. Which function is used to concatenate two C-style strings?

a) strcat()
b) concat()
c) append()
d) join()

Answer: a) strcat()

92. What is the correct way to compare two C-style strings in C++?

a) strcompare()
b) strcmp()
c) compare()
d) stringCompare()

Answer: b) strcmp()

93. Which of the following statements is true about arrays in C++?

a) Arrays can only store elements of the same data type.
b) Arrays can dynamically resize during runtime.
c) The size of an array must be known at compile time.
d) Arrays are not supported in C++.

Answer: c) The size of an array must be known at compile time.

94. How do you initialize an array in C++?

a) array = {1, 2, 3};
b) int array[3] = {1, 2, 3};
c) array(1, 2, 3);
d) array = new int[3];

Answer: b) int array[3] = {1, 2, 3};

95. What is the output of the following code?

```
int numbers[] = {1, 2, 3, 4, 5};
cout << numbers[2];
```

a) 2
b) 3
c) 4
d) 5

Answer: b) 3

96. What does the sizeof operator in C++ return for an array?

a) The number of elements in the array.
b) The sum of all elements in the array.
c) The total size (in bytes) of the array.
d) The average value of elements in the array.

Answer: c) The total size (in bytes) of the array.

97. What is the incorrect way to declare and initialize a character array in C++?

a) char name = "John";
b) char name[] = "John";
c) string name = "John";
d) string name[] = {'J', 'o', 'h', 'n'};

Answer: a) char name = "John";

98. How do you find the length of a C++ string object?

a) length() method
b) size() method
c) strlen() function
d) sizeOf() function

Answer: b) size() method

99. Which function is used to copy one C-style string into another?

a) strcpy()
b) copy()
c) strncpy()
d) stringCopy()

Answer: a) strcpy()

100. What is the purpose of the getline() function in C++ when used with strings?

a) To read a line of text from the console.
b) To concatenate two strings.
c) To find the length of a string.
d) To compare two strings.

Answer: a) To read a line of text from the console.

101.If an algorithm has a time complexity of O(1), what does it imply?

a) It runs in constant time.
b) It runs in logarithmic time.

c) It runs in linear time.
d) It runs in polynomial time.

Answer: a) It runs in constant time.

102. What is the time complexity of a linear search algorithm in an array of size n?

a) O(1)
b) O(n)
c) O(log n)
d) O(n^2)

Answer: b) O(n)

103. If an algorithm has a time complexity of O(log n), what type of algorithm is it likely to be?

a) Linear algorithm
b) Quadratic algorithm
c) Exponential algorithm
d) Logarithmic algorithm

Answer: d) Logarithmic algorithm

105. What is the space complexity of a recursive algorithm with a depth of recursion log n?
a) O(1)
b) O(n)
c) O(log n)
d) O(n^2)

Answer: c) O(log n)

106. If an algorithm creates a matrix of size n x n, what is its space complexity?

a) O(1)
b) O(n)
c) O(n^2)
d) O(log n)

Answer: c) O(n^2)

107. In binary search, what is the key requirement for the array?
a) It must be sorted in descending order.
b) It must be sorted in ascending order.
c) It can be in any order.
d) It must have only unique elements.

Answer: b) It must be sorted in ascending order.

108. What is the time complexity of binary search on a unsorted array of size n?

a) O(1)
b) O(log n)
c) O(n)
d) not possible

Answer: d) not possible

109. If the target element is not present in the array during binary search, what does the algorithm return?

a) -1
b) 0
c) The index of the last element
d) An error message

Answer: a) -1

110. Which of the following is a disadvantage of recursive binary search?

a) Simplicity of implementation
b) Stack overflow for large arrays
c) Efficient for unsorted arrays
d) Suitable for dynamic arrays only

Answer: b) Stack overflow for large arrays

111. What is the efficient formula to calculate the mid-point index in binary search for large arrays?

a) mid = (low + high) / 2
b) mid = (low - high) / 2
c) mid = low + (high - low) / 2
d) mid = low * high

Answer: c) mid = low + (high - low) / 2

112. Which algorithm is often used as a base for binary search?

a) Linear Search
b) Bubble Sort
c) Quick Sort
d) Merge Sort

Answer: d) Merge Sort

113. In binary search, what is the purpose of checking the middle element against the target?

a) To find the index of the target element.
b) To determine if the array is sorted.
c) To decide whether to search the left or right subarray.
d) To reorder the array.

Answer: c) To decide whether to search the left or right subarray.

114. What is the advantage of binary search over linear search?

a) Binary search works for unsorted arrays.
b) Binary search has a lower time complexity.
c) Binary search is easier to implement.
d) Binary search works only for small arrays.

Answer: b) Binary search has a lower time complexity.

115. What is bitmasking commonly used for in programming?

a) Text processing
b) Mathematical calculations
c) Manipulating individual bits in variables
d) String manipulation

Answer: c) Manipulating individual bits in variables

116. In bitwise AND operation (&), what is the result if both bits are 1?

a) 0
b) 1
c) -1
d) No change

Answer: b) 1

117. What is the result of the expression 1 << 3?

a) 8
b) 4
c) 2
d) 1

Answer: a) 8

118. Which bitwise operation is used to set a particular bit to 1?

a) AND (&)
b) OR (|)
c) XOR (^)
d) NOT (~)

Answer: b) OR (|)

119. In bitwise XOR operation (^), what is the result if both bits are the same?
a) 0
b) 1
c) -1
d) No change

Answer: a) 0

120. What is the purpose of the << operator in bitmasking?

a) Left shift
b) Right shift
c) Bitwise AND
d) Bitwise OR

Answer: a) Left shift

121. Which bitwise operation is used to toggle a particular bit?

a) AND (&)
b) OR (|)
c) XOR (^)
d) NOT (~)

Answer: c) XOR (^)

122. What is the result of the expression 5 & 3?

a) 0
b) 1
c) 3
d) 5

Answer: c) 3

123. What is backtracking?

a) A search algorithm
b) An optimization technique
c) A technique to explore all possible solutions
d) A data structure

Answer: c) A technique to explore all possible solutions

124. In backtracking, when is pruning typically done?

a) Before exploring any solution
b) After exploring all solutions
c) During the exploration of solutions
d) Pruning is not used in backtracking

Answer: c) During the exploration of solutions

125. What is the key characteristic of a problem suitable for backtracking?

a) The problem has a unique solution.
b) The problem can be broken into subproblems.
c) The problem is solved using a loop.
d) The problem involves sorting.
.

Answer: a) It represents all possible choices at each decision point.

127. Which of the following problems is well-suited for backtracking?

a) Linear search
b) subset problems
c) Sudoku solving
d) Binary search

Answer: c) Sudoku solving

128. What does the term "backtrack" refer to in the context of backtracking algorithms?

a) Going backward in time
b) Undoing the last decision and trying a different one
c) Iterating over all choices before making a decision
d) Repeating the same decision

Answer: b) Undoing the last decision and trying a different one

129. In backtracking, what is a "partial solution"?

a) The final solution to the problem.
b) An incomplete solution that may be expanded further.

c) The first solution generated during exploration.
d) A solution obtained by pruning choices.

Answer: b) An incomplete solution that may be expanded further.

130. Which data structure is commonly used for keeping track of choices in backtracking?

a) Stack
b) Queue
c) Linked list
d) Array

Answer: a) Stack

131. What is the purpose of Big-O notation in computer science?

a) To represent the worst-case time complexity of an algorithm.
b) To represent the average-case time complexity of an algorithm.
c) To represent the best-case time complexity of an algorithm.
d) To represent the space complexity of an algorithm.

Answer: a) To represent the worst-case time complexity of an algorithm.

132. What does $\Omega(n \log n)$ represent in asymptotic notation?

a) Best-case time complexity
b) Average-case time complexity
c) Worst-case time complexity
d) Lower bound time complexity

Answer: d) Lower bound time complexity

133. If an algorithm has a time complexity of $O(1)$, what can be said about its efficiency?

a) It is very efficient.
b) It is moderately efficient.
c) It is less efficient.
d) It depends on other factors.

Answer: a) It is very efficient.

134. Which of the following statements is true about the relationship between O(f(n)) and Θ(f(n))?

a) O(f(n)) always equals Θ(f(n)).
b) O(f(n)) is a subset of Θ(f(n)).
c) O(f(n)) is a superset of Θ(f(n)).
d) O(f(n)) and Θ(f(n)) are unrelated.

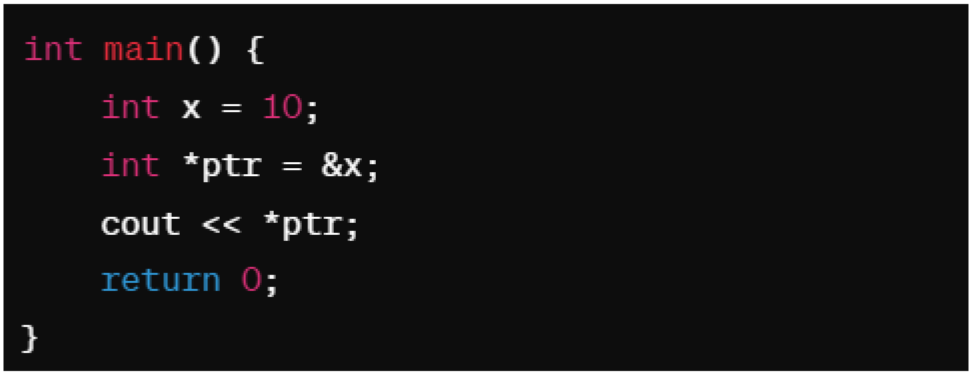
Answer: b) O(f(n)) is a subset of Θ(f(n)).

135. What is the significance of the term "asymptotic" in asymptotic notation?
a) It represents the average case.
b) It represents the best case.
c) It represents the worst case.
d) It describes the behavior as the input approaches infinity.
Answer: d) It describes the behavior as the input approaches infinity.

What is a pointer in C++?
a) A variable that stores the memory address of another variable
b) A variable that stores integer values only
c) A variable that can be accessed globally
d) A reserved keyword in C++

Answer: a) A variable that stores the memory address of another variable

What is the output of the following code ?



```
int main() {
    int x = 10;
    int *ptr = &x;
    cout << *ptr;
    return 0;
}
```

a) 10
b) Memory address of x
c) Compilation error
d) Garbage value

Answer: a) 10

What does the dereference operator (*) do in C++?

a) It declares a pointer variable
b) It assigns a value to a pointer
c) It accesses the value pointed to by a pointer
d) It compares two pointers

Answer: c) It accesses the value pointed to by a pointer

What is the size of a pointer in C++ (on a 64-bit system)?
a) 4 bytes
b) 8 bytes
c) Depends on the data type it points to
d) Depends on the compiler

Answer: b) 8 bytes

What is the null pointer in C++?
a) A pointer with value 0
b) A pointer with value -1
c) A pointer with value 1
d) A pointer with uninitialized value

Answer: a) A pointer with value 0

What is the purpose of the 'new' operator in C++?

a) It deallocates memory
b) It creates a new variable
c) It dynamically allocates memory on the heap
d) It is used for pointer arithmetic

Answer: c) It dynamically allocates memory on the heap

What is a dangling pointer in C++?

a) A pointer that points to a valid memory address
b) A pointer that points to an invalid or deallocated memory address
c) A pointer that is uninitialized
d) A pointer that is declared but not used

Answer: b) A pointer that points to an invalid or deallocated memory address

What is the purpose of the 'delete' operator in C++?

a) It deletes a variable from memory
b) It deallocates memory allocated by 'new'
c) It assigns a value to a pointer
d) It initializes a pointer

Answer: b) It deallocates memory allocated by 'new'

What is a pointer to a function in C++?

a) A pointer that stores the address of a variable
b) A pointer that stores the address of another function
c) A pointer that stores a function's return value
d) A pointer that stores a function's arguments

Answer: b) A pointer that stores the address of another function

What is the purpose of the 'nullptr' keyword in modern C++?

a) It is equivalent to the null pointer (NULL)
b) It is used for pointer arithmetic
c) It represents a pointer to a specific memory address
d) It is used to initialize pointers to point to nothing

Answer: d) It is used to initialize pointers to point to nothing