CS301
Data Structures

Important mcqs

Lec 1 - Introduction to Data Structures

1. **Which data structure follows the "last-in-first-out" (LIFO) principle?** a. Queue b. Stack c. Linked List d. Tree Solution: b

2. **Which data structure allows for efficient insertion and deletion operations in the middle?** a. Array b. Stack c. Queue d. Linked List Solution: d

3. **Which data structure is used to represent hierarchical relationships?** a. Array b. Stack c. Queue d. Tree Solution: d

4. **Which data structure is used to find the shortest path between two nodes in a network?** a. Array b. Stack c. Queue d. Graph Solution: d

5. **Which data structure stores elements in a non-linear and hierarchical manner?** a. Array b. Stack c. Queue d. Tree Solution: d

6. **Which data structure follows the "first-in-first-out" (FIFO) principle?** a. Queue b. Stack c. Linked List d. Tree Solution: a

7. **Which data structure is used to reverse the order of elements in a sequence?** a. Array b. Stack c. Queue d. Linked List Solution: b

8. **Which data structure is used to implement a symbol table?** a. Array b. Stack c. Queue d. Hash table Solution: d

9. **Which data structure is used to sort elements in a sequence?** a. Array b. Stack c. Queue d. Heap Solution: d

10. **Which data structure is used to store and access elements based on a key-value pair?** a. Array b. Stack c. Queue d. Dictionary Solution: d

Lec 2 - List Implementation

1. **Which data structure is used to implement a list?**
   a) Array
   b) Stack
   c) Queue
   d) Tree
   Answer: a) Array

2. **Which of the following is a disadvantage of array-based lists?**
   a) Efficient random access to elements
   b) Dynamic resizing of the list
   c) Inefficient insertion and deletion operations
   d) No need to allocate memory in advance
   Answer: c) Inefficient insertion and deletion operations

3. **Which of the following is a disadvantage of linked-list based lists?**
   a) Efficient random access to elements
   b) Dynamic resizing of the list
   c) Inefficient insertion and deletion operations
   d) No need to allocate memory in advance
   Answer: a) Efficient random access to elements

4. **Which of the following is true for an empty list?**
   a) It has no elements
   b) It has one element
   c) It has multiple elements
   d) None of the above
   Answer: a) It has no elements

5. **Which operation adds an element at the end of an array-based list?**
   a) push()
   b) pop()
   c) insert()
   d) append()
   Answer: d) append()

6. **Which operation adds an element at the beginning of a linked-list based list?**
   a) push()
   b) pop()
   c) insert()
   d) append()
   Answer: a) push()

7. **Which operation removes the last element from an array-based list?**
   a) push()
   b) pop()
   c) insert()
   d) append()
   Answer: b) pop()

8. **Which operation removes the first element from a linked-list based list?**
   a) push()
   b) pop()
   c) insert()
   d) append()
   Answer: b) pop()

9. **Which of the following is a common application of lists?**
   a) Implementing a queue
   b) Implementing a hash table
   c) Implementing a binary search tree
   d) Implementing a heap
   Answer: a) Implementing a queue

10. **Which of the following is not a type of list?**
    a) Singly linked list
    b) Doubly linked list
    c) Circular linked list
    d) Heap linked list
    Answer: d) Heap linked list

Lec 3 - Linked List inside Computer Memory

1. **In a linked list, each node contains:**
   a. A value and a pointer to the previous node
   b. A value and a pointer to the next node
   c. A key and a value
   d. A key and a pointer to the next node
   Answer: b

2. **The first node in a linked list is called the:**
   a. Head
   b. Tail
   c. Root
   d. Leaf
   Answer: a

3. **In computer memory, each node in a linked list is typically represented as:**
   a. A block of memory that contains the value and a pointer to the previous node
   b. A block of memory that contains the value and a pointer to the next node
   c. A hash table that contains the key and the value
   d. An array that contains the key and a pointer to the next node
   Answer: b

4. **What is the time complexity of inserting a node at the beginning of a linked list?**
   a. O(1)
   b. O(n)
   c. O(log n)
   d. O(n log n)
   Answer: a

5. **What is the time complexity of inserting a node at the end of a linked list?**
   a. O(1)
   b. O(n)
   c. O(log n)
   d. O(n log n)
   Answer: b

6. **Deleting a node from a linked list requires updating the:**
   a. Previous node's pointer to the next node
   b. Next node's pointer to the previous node
   c. Current node's value to NULL
   d. None of the above
   Answer: a

7. **Traversing a linked list means:**
   a. Deleting a node from the list
   b. Inserting a node into the list
   c. Moving through the list from the head to the tail
   d. Sorting the list in ascending order
   Answer: c

8. **Which of the following is a disadvantage of linked lists compared to arrays?**
   a. Linked lists allow for efficient insertion and deletion of nodes
   b. Linked lists use memory flexibly
   c. Linked lists can grow dynamically
   d. Linked lists have slow access times for specific nodes
   Answer: d

9. **Which of the following operations can be performed in constant time on a linked list?**
   a. Finding the maximum value in the list
   b. Inserting a node at the end of the list
   c. Removing the head node from the list
   d. Sorting the list in descending order
   Answer: c

10. **What is the space complexity of a linked list?**
    a. O(n)
    b. O(log n)
    c. O(1)
    d. O(n log n)
    Answer: a

Lec 4 - Methods of Linked List

1. **What is the time complexity of inserting an element at the beginning of a singly linked list?**
   a) O(n)
   b) O(log n)
   c) O(1)
   d) O(n^2)
   Answer: c) O(1)

2. **Which of the following is not a type of Linked List?**
   a) Singly Linked List
   b) Doubly Linked List
   c) Circular Linked List
   d) Binary Linked List
   Answer: d) Binary Linked List

3. **What is the time complexity of inserting an element at the end of a singly linked list?**
   a) O(n)
   b) O(log n)
   c) O(1)
   d) O(n^2)
   Answer: a) O(n)

4. **Which of the following is not a pointer used in Linked List?**
   a) head pointer
   b) tail pointer
   c) node pointer
   d) root pointer
   Answer: d) root pointer

5. **What is the time complexity of deleting an element from a singly linked list?**
   a) O(n)
   b) O(log n)
   c) O(1)
   d) O(n^2)
   Answer: a) O(n)

6. **Which of the following Linked List traversal technique involves recursion?**
   a) Linear traversal
   b) Binary traversal
   c) Depth First Traversal
   d) Breadth First Traversal
   Answer: c) Depth First Traversal

7. Which of the following is a disadvantage of using a doubly linked list?
   a) Faster traversal in both directions
   b) Requires more memory than singly linked list
   c) More difficult to implement than singly linked list
   d) Allows insertion and deletion of elements only at the beginning of the list
   Answer: b) Requires more memory than singly linked list

8. **Which of the following is not a type of node used in Linked List?**
   a) data node
   b) header node
   c) sentinel node
   d) tail node
   Answer: a) data node

9. **Which of the following Linked List method is used to reverse the order of elements in the list?**
   a) reverse()
   b) rotate()
   c) shuffle()
   d) sort()
   Answer: a) reverse()

10. **Which of the following is a type of circular linked list?**
    a) Circular doubly linked list
    b) Circular binary linked list
    c) Circular balanced linked list
    d) Circular heap linked list
    Answer: a) Circular doubly linked list

Lec 5 - Benefits of using circular list

1. **What is a benefit of using a circular linked list?**
   a) It has a fixed size
   b) It allows for efficient implementation of circular structures
   c) It cannot be traversed multiple times
   d) It is less efficient than a linear linked list

Answer: b

2. **What is a circular buffer?**
   a) A data structure that can only be accessed in a circular order
   b) A data structure that can only be accessed in a linear order
   c) A buffer that can be efficiently implemented using a circular linked list
   d) A buffer that can only hold a fixed number of elements

Answer: c

3. **How can a circular linked list simplify insertion at the beginning or end?**
   a) It requires more memory than a linear linked list
   b) It requires less memory than a linear linked list
   c) It requires the same amount of memory as a linear linked list
   d) It cannot simplify insertion at the beginning or end

Answer: b

4. **What is a disadvantage of using a circular linked list?**
   a) It is less efficient than a linear linked list
   b) It cannot represent circular structures
   c) It is more difficult to implement than a linear linked list
   d) It has a fixed size

Answer: a

5. **Which algorithm can be efficiently implemented using a circular linked list?**
   a) Binary search
   b) Depth-first search
   c) Breadth-first search
   d) Traversing the list multiple times

Answer: d

6. **What is a circular linked list?**
   a) A linked list where each node has a pointer to the previous node
   b) A linked list where each node has a pointer to the next node and the previous node
   c) A linked list where the last node points to the first node
   d) A linked list where the first node points to the last node

Answer: c

7. **What is a benefit of using a circular linked list for representing a clock?**
   a) It is less efficient than a linear linked list
   b) It allows for efficient implementation of circular structures
   c) It requires more memory than a linear linked list
   d) It cannot represent circular structures

Answer: b

8. **What is a circular linked list used for?**
   a) Representing trees
   b) Implementing binary search
   c) Implementing circular buffering
   d) Storing fixed-size data

Answer: c

9. **Can a circular linked list be traversed multiple times?**
   a) Yes, but it is less efficient than a linear linked list
   b) No, it can only be traversed once
   c) Yes, and it is more efficient than a linear linked list
   d) Yes, but it requires more memory than a linear linked list

Answer: c

10. **How is deletion at the end of a circular linked list implemented?**
    a) The last node's pointer is set to NULL
    b) The last node's pointer is set to the first node
    c) The second to last node's pointer is set to NULL
    d) The second to last node's pointer is set to the first node

Answer: c

Lec 6 - Stack From the Previous Lecture

1. **Which of the following is NOT a characteristic of a stack?**
   a. Follows LIFO principle
   b. Has two main operations: push and pop
   c. Can only be implemented using arrays
   d. Topmost element is the last one added

Answer: c. Can only be implemented using arrays

2. **What is the time complexity of push and pop operations in a stack implemented using an array?**
   a. O(1)
   b. O(log n)
   c. O(n)
   d. O(n^2)

Answer: a. O(1)

3. **Which data structure is often used to implement a stack?**
   a. Array
   b. Linked List
   c. Queue
   d. Binary Tree

Answer: b. Linked List

4. **Which of the following is NOT a common application of stacks?**
   a. Reversing a string
   b. Evaluating postfix expressions
   c. Implementing depth-first search in a graph
   d. Sorting an array

Answer: d. Sorting an array

5. **Which operation in a stack does not modify the stack?**
   a. Push
   b. Pop
   c. Peek
   d. Size

Answer: c. Peek

6. **What happens when we try to pop an element from an empty stack?**
   a. The program crashes
   b. An error message is displayed
   c. The topmost element becomes NULL
   d. Nothing happens

Answer: b. An error message is displayed

7. **Which of the following is NOT a disadvantage of using an array to implement a stack?**
   a. Fixed size
   b. Elements must be contiguous in memory
   c. Dynamic resizing is difficult
   d. Push operation is slower than pop operation

Answer: d. Push operation is slower than pop operation

8. **What is the maximum number of elements a stack implemented using an array can hold if its size is n?**
   a. n
   b. n-1
   c. 2n
   d. 2n-1

Answer: b. n-1

9. **Which of the following is NOT a potential application of stacks in computer science?**
   a. Function calls and return values
   b. Undo and redo operations
   c. Implementing breadth-first search in a graph
   d. Checking for balanced parentheses in an expression

Answer: c. Implementing breadth-first search in a graph

10. **What is the time complexity of searching for an element in a stack implemented using a linked list?**
    a. O(1)
    b. O(log n)
    c. O(n)
    d. O(n^2)

Answer: c. O(n)

Lec 7 - Evaluating postfix expressions

1. **What is postfix notation?**
   a) Operators are written before their operands
   b) Operators are written after their operands
   c) Operators are written in between their operands
   d) None of the above

Answer: b) Operators are written after their operands

2. **Which data structure is used to evaluate postfix expressions?**
   a) Queue
   b) Linked List
   c) Stack
   d) Tree

Answer: c) Stack

3. **Which of the following is an example of a postfix expression?**
   a) 2 + 3
   b) 2 * 3 + 4
   c) 2 3 +
   d) (2 + 3) * 4

Answer: c) 2 3 +

4. **What is the first step in evaluating a postfix expression?**
   a) Scan the expression from right to left
   b) Scan the expression from left to right
   c) Push the first operand onto the stack
   d) Pop the first operand from the stack

Answer: b) Scan the expression from left to right

5. **What happens when an operand is encountered in a postfix expression?**
   a) It is pushed onto the stack
   b) It is popped from the stack
   c) It is ignored
   d) None of the above

Answer: a) It is pushed onto the stack

6. **What happens when an operator is encountered in a postfix expression?**
   a) It is pushed onto the stack
   b) It is popped from the stack
   c) The top two operands are popped from the stack, and the operation is performed on them
   d) None of the above

Answer: c) The top two operands are popped from the stack, and the operation is performed on them

7. **Which of the following is an example of a postfix expression that involves multiplication?**
   a) 2 + 3
   b) 2 * 3 + 4
   c) 2 3 *
   d) (2 + 3) * 4

Answer: c) 2 3 *

8. **What happens when the entire postfix expression has been scanned?**
   a) The result is the top element in the stack
   b) The result is the bottom element in the stack
   c) The stack is empty
   d) None of the above

Answer: a) The result is the top element in the stack

9. **Which of the following is an example of a postfix expression that involves subtraction?**
   a) 2 + 3
   b) 2 * 3 + 4
   c) 2 3 -
   d) (2 + 3) * 4

Answer: c) 2 3 -

10. **Which of the following is an example of a postfix expression that involves division?**
    a) 2 + 3
    b) 2 * 3 + 4
    c) 2 3 /
    d) (2 + 3) * 4

Answer: c) 2 3 /

Lec 8 - Conversion from infix to postfix

1. **Which of the following data structures is commonly used for converting infix expressions to postfix expressions?**
   a. Stack
   b. Queue
   c. Linked List
   d. Heap
   Answer: a. Stack

2. **In infix notation, where are operators written in relation to their operands?**
   a. Before
   b. After
   c. Between
   d. Both a and c
   Answer: d. Both a and c

3. **What is the first step in converting an infix expression to postfix notation?**
   a. Scan the expression from left to right
   b. Initialize an empty stack and postfix expression
   c. Check for balanced parentheses
   d. Add operands to the postfix expression
   Answer: b. Initialize an empty stack and postfix expression

4. **Which of the following operators has the highest precedence in mathematical expressions?**
   a. Multiplication
   b. Division
   c. Addition
   d. Subtraction
   Answer: a. Multiplication

5. **What is the role of a stack in converting infix to postfix notation?**
   a. To keep track of operators and their precedence levels
   b. To keep track of operands and their positions
   c. To perform the necessary calculations
   d. To check for errors in the expression
   Answer: a. To keep track of operators and their precedence levels

6. **What happens to a left parenthesis when converting from infix to postfix notation?**
   a. It is added to the postfix expression
   b. It is pushed onto the stack
   c. It is discarded
   d. It is popped off the stack
   Answer: b. It is pushed onto the stack

7. **How can errors be handled while converting infix to postfix notation?**
   a. By checking for balanced parentheses
   b. By checking for errors during scanning
   c. By using a queue data structure
   d. Both a and b
   Answer: d. Both a and b

8. **Which of the following expressions is equivalent to "a + b * c" in postfix notation?**
   a. "a b c * +"
   b. "a b + c *"
   c. "a + b c *"
   d. "a b c + *"
   Answer: a. "a b c * +"

9. **What is the final step in converting an infix expression to postfix notation?**
   a. Pop any remaining operators off the stack and add them to the postfix expression
   b. Add operands to the postfix expression
   c. Discard the left parenthesis
   d. Push the right parenthesis onto the stack
   Answer: a. Pop any remaining operators off the stack and add them to the postfix expression

10. **Which of the following is an advantage of postfix notation over infix notation?**
    a. It is easier to read and write
    b. It eliminates the need for parentheses to indicate the order of operations
    c. It is more commonly used in mathematical expressions
    d. Both a and b
    Answer: b. It eliminates the need for parentheses to indicate the order of operations.

Lec 9 - Memory Organization

1. **Which of the following is the fastest type of memory?**
   a) Registers
   b) Cache
   c) Main memory
   d) Secondary storage
   Answer: a) Registers

2. **Which level of memory is located close to the CPU and used to store frequently accessed data?**
   a) Registers
   b) Cache
   c) Main memory
   d) Secondary storage
   Answer: b) Cache

3. **Which level of memory is used to store data and program instructions during execution?**
   a) Registers
   b) Cache
   c) Main memory
   d) Secondary storage
   Answer: c) Main memory

4. **Which type of memory has the largest capacity?**
   a) Registers
   b) Cache
   c) Main memory
   d) Secondary storage
   Answer: d) Secondary storage

5. **Which type of storage is used for long-term data storage?**
   a) Main memory
   b) Secondary storage
   c) Cache
   d) Registers
   Answer: b) Secondary storage

6. **Which level of memory has the slowest access time?**
   a) Registers
   b) Cache
   c) Main memory
   d) Secondary storage
   Answer: d) Secondary storage

7. **Which level of memory is the most expensive?**
   a) Registers
   b) Cache
   c) Main memory
   d) Secondary storage
   Answer: a) Registers

8. **Which level of memory has the largest capacity?**
   a) Registers
   b) Cache
   c) Main memory
   d) Secondary storage
   Answer: d) Secondary storage

9. **Which type of memory organization is used to minimize the time spent accessing data from higher levels of the memory hierarchy?**
   a) Top-down organization
   b) Bottom-up organization
   c) Parallel organization
   d) Hierarchical organization
   Answer: d) Hierarchical organization

10. **Which type of computing system may use different types of memory organization depending on its intended use and performance requirements?**
    a) Embedded systems
    b) Personal computers
    c) Supercomputers
    d) All of the above
    Answer: d) All of the above

Lec 10 - Queues

1. **What is a queue data structure?**
   a) A data structure where the last element added is the first one to be removed.
   b) A data structure where the first element added is the first one to be removed.
   c) A data structure where the middle element is always removed first.
   d) None of the above.

Answer: b) A data structure where the first element added is the first one to be removed.

2. **Which operation adds an element to the queue?**
   a) Dequeue
   b) Enqueue
   c) Peek
   d) None of the above.

Answer: b) Enqueue

3. **Which operation removes an element from the queue?**
   a) Dequeue
   b) Enqueue
   c) Peek
   d) None of the above.

Answer: a) Dequeue

4. **Which operation returns the element at the front of the queue without removing it?**
   a) Dequeue
   b) Enqueue
   c) Peek
   d) None of the above.

Answer: c) Peek

5. **Which data structure is commonly used to implement a queue?**
   a) Array
   b) Linked list
   c) Both a and b
   d) None of the above.

Answer: c) Both a and b

6. **Which of the following is not a real-world application of queues?**
   a) Waiting lines at banks
   b) Amusement parks
   c) Airports
   d) None of the above.

Answer: d) None of the above.

7. **What is the time complexity of the enqueue operation in a queue implemented using an array?**
   a) O(1)
   b) O(n)
   c) O(log n)
   d) None of the above.

Answer: a) O(1)

8. **What is the time complexity of the dequeue operation in a queue implemented using a linked list?**
   a) O(1)
   b) O(n)
   c) O(log n)
   d) None of the above.

Answer: a) O(1)

9. **Which of the following is a disadvantage of using an array to implement a queue?**
   a) Insertion and deletion are faster than in a linked list.
   b) The size of the array must be fixed.
   c) It is more efficient in terms of memory usage.
   d) None of the above.

Answer: b) The size of the array must be fixed.

10. **Which of the following is a disadvantage of using a linked list to implement a queue?**
    a) Insertion and deletion are slower than in an array.
    b) The size of the linked list must be fixed.
    c) It is less efficient in terms of memory usage.
    d) None of the above.

Answer: a) Insertion and deletion are slower than in an array.

Lec 11 - Implementation of Priority Queue

1. **Which of the following data structures is commonly used to implement a Priority Queue?**
   a) Linked List
   b) Queue
   c) Binary Heap
   d) Stack
   Solution: c) Binary Heap

2. **Which of the following operations is commonly supported by a Priority Queue?**
   a) Enqueue
   b) Dequeue
   c) Insert
   d) All of the above
   Solution: d) All of the above

3. **What is the time complexity of inserting an element into a binary heap?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) O(n log n)
   Solution: b) O(log n)

4. **What is the time complexity of deleting the highest-priority element from a binary heap?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) O(n log n)
   Solution: b) O(log n)

5. **Which of the following is an advantage of using a Fibonacci Heap to implement a Priority Queue?**
   a) Faster insert operation than a binary heap
   b) Lower memory usage than a binary heap
   c) Faster delete operation than a binary heap
   d) All of the above
   Solution: c) Faster delete operation than a binary heap

6. **Which of the following algorithms makes use of a Priority Queue?**
   a) Dijkstra's shortest path algorithm
   b) Binary search algorithm
   c) Bubble sort algorithm
   d) Linear search algorithm
   Solution: a) Dijkstra's shortest path algorithm

7. **Which of the following data structures is commonly used to implement a Priority Queue in C++?**
   a) std::queue
   b) std::vector
   c) std::list
   d) std::priority_queue
   Solution: d) std::priority_queue

8. **Which of the following operations is not supported by a Priority Queue?**
   a) Changing the priority of an element
   b) Inserting an element
   c) Removing an element with the lowest priority
   d) Removing an element with the highest priority
   Solution: c) Removing an element with the lowest priority

9. **Which of the following is an application of Priority Queues?**
   a) Sorting large datasets
   b) Implementing a stack
   c) Implementing a queue
   d) Task scheduling in an operating system
   Solution: d) Task scheduling in an operating system

10. **Which of the following is a disadvantage of using a binary heap to implement a Priority Queue?**
    a) Slower delete operation than a Fibonacci Heap
    b) Higher memory usage than a Fibonacci Heap
    c) Slower insert operation than a Fibonacci Heap
    d) All of the above
    Solution: a) Slower delete operation than a Fibonacci Heap

Lec 12 - Operations on Binary Tree

1. **What is the height of a binary tree with only one node?**
   a) 0
   b) 1
   c) 2
   d) Undefined

Answer: a) 0

2. **Which traversal method visits the nodes in the order left subtree, root, right subtree?**
   a) Pre-order traversal
   b) In-order traversal
   c) Post-order traversal
   d) Level-order traversal

Answer: b) In-order traversal

3. **Which traversal method visits the nodes in the order root, left subtree, right subtree?**
   a) Pre-order traversal
   b) In-order traversal
   c) Post-order traversal
   d) Level-order traversal

Answer: a) Pre-order traversal

4. **What is the time complexity of searching for a node in a Binary Tree?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) It depends on the implementation

Answer: c) O(n)

5. **What is the maximum number of nodes at level k in a Binary Tree?**
   a) 2^k
   b) k^2
   c) k+1
   d) None of the above

Answer: a) 2^k

6. **Which of the following is a way to delete a node in a Binary Tree?**
   a) Deleting the node and its children
   b) Replacing the node with its left child
   c) Replacing the node with its right child
   d) All of the above

Answer: d) All of the above

7. **What is the time complexity of finding the height of a Binary Tree?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) It depends on the implementation

Answer: c) O(n)

8. **What is the maximum number of nodes in a Binary Tree with height h?**
   a) 2^h
   b) h^2
   c) h+1
   d) None of the above

Answer: a) 2^h - 1

9. **Which of the following is a way to insert a node in a Binary Tree?**
   a) As the left child of a leaf node
   b) As the right child of a leaf node
   c) As the left child of a non-leaf node
   d) All of the above

Answer: d) All of the above

10. **Which of the following is an advantage of using a Binary Tree over a linked list?**
    a) Binary Tree can be searched faster than a linked list
    b) Binary Tree can be sorted faster than a linked list
    c) Binary Tree can store data in a hierarchical structure
    d) All of the above

Answer: d) All of the above

Lec 13 - Cost of Search

1. **What is the cost of search in computer science?**

a. Amount of time, resources, and computational power required to search for a specific item in a data structure
b. The amount of data in a data structure
c. The size of the data structure
d. The type of data structure used

Answer: a.

2. **Which of the following measures is a common measure of search cost?**

a. Time complexity
b. Space complexity
c. Worst-case analysis
d. All of the above

Answer: d.

3. **Which search algorithm is commonly used for searching a sorted array?**

a. Binary search
b. Linear search
c. Depth-first search
d. Breadth-first search

Answer: a.

4. **Which data structure is commonly used for implementing hash tables?**

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

Answer: a.

5. **What is the time complexity of binary search?**

a. O(1)
b. O(log n)
c. O(n)
d. O(n log n)

Answer: b.

6. **Which of the following is a disadvantage of linear search?**

a. It has a time complexity of O(1)
b. It is easy to implement
c. It has a time complexity of O(n)
d. It is efficient for large datasets

Answer: c.

7. **Which of the following is a disadvantage of binary search?**

a. It can only be used with sorted arrays
b. It has a time complexity of O(n)
c. It is difficult to implement
d. It is not efficient for large datasets

Answer: a.

8. **Which data structure is commonly used for implementing binary search trees?**

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

Answer: b.

9. **What is the time complexity of searching a hash table with a good hash function?**

a. O(1)
b. O(log n)
c. O(n)
d. O(n log n)

Answer: a.

10. **Which of the following measures is not a common measure of search cost?**

a. Time efficiency
b. Space efficiency
c. Worst-case analysis
d. Best-case analysis

Answer: d.

Lec 14 - Recursive Calls

1. **What is recursion?**
   a. A process of dividing a problem into smaller subproblems
   b. A process of repeating a set of instructions
   c. A process of sorting data in ascending order
   d. None of the above

Answer: a

2. **What is the base case in recursion?**
   a. The case where the function calls itself
   b. The case where the function returns a value without calling itself
   c. The case where the function uses a loop instead of recursion
   d. None of the above

Answer: b

3. **Which data structure is commonly used in recursion?**
   a. Stack
   b. Queue
   c. Linked list
   d. Array

Answer: a

4. **What is the maximum number of recursive calls that can be made?**
   a. 100
   b. 1000
   c. It depends on the available memory
   d. There is no limit

Answer: c

5. **Which of the following is a disadvantage of recursion?**
   a. It is easy to understand and implement
   b. It may cause stack overflow errors
   c. It always results in better performance than iterative solutions
   d. None of the above

Answer: b

6. **Which of the following algorithms uses recursion?**
   a. Quick sort
   b. Merge sort
   c. Bubble sort
   d. Selection sort

Answer: a and b

7. **Which of the following is true about recursive functions?**
   a. They are always faster than iterative functions
   b. They require less memory than iterative functions
   c. They may be more readable and concise than iterative functions
   d. None of the above

Answer: c

8. **What is tail recursion?**
   a. A type of recursion where the recursive call is the last operation performed by the function
   b. A type of recursion where the function calls itself multiple times
   c. A type of recursion where the function uses a loop instead of recursion
   d. None of the above

Answer: a

9. **What is the difference between direct and indirect recursion?**
   a. Direct recursion occurs when a function calls itself, while indirect recursion occurs when two or more functions call each other.
   b. Direct recursion occurs when two or more functions call each other, while indirect recursion occurs when a function calls itself.
   c. Direct and indirect recursion are the same thing.
   d. None of the above

Answer: a

10. **Which of the following is an example of a recursive data structure?**
    a. Linked list
    b. Stack
    c. Queue
    d. Array

Answer: a and b

Lec 15 - Level-order Traversal of a Binary Tree

1. **What is level-order traversal of a binary tree?**
   A. Visiting the root node first
   B. Visiting the nodes level by level
   C. Visiting the left child first
   D. Visiting the right child first
   Answer: B

2. **Which data structure can be used to implement level-order traversal?**
   A. Stack
   B. Queue
   C. Linked list
   D. Binary search tree
   Answer: B

3. **What is the time complexity of level-order traversal?**
   A. O(log n)
   B. O(n)
   C. O(n^2)
   D. O(2^n)
   Answer: B

4. **In which order are the nodes visited in level-order traversal?**
   A. Left to right, bottom to top
   B. Right to left, top to bottom
   C. Left to right, top to bottom
   D. Right to left, bottom to top
   Answer: C

5. **Which traversal technique can be used to print the nodes of a binary tree in level-order?**
   A. In-order traversal
   B. Post-order traversal
   C. Pre-order traversal
   D. Level-order traversal
   Answer: D

6. **What is the space complexity of level-order traversal?**
   A. O(1)
   B. O(n)
   C. O(n^2)
   D. O(2^n)
   Answer: B

7. **Level-order traversal can be used to solve which type of problem?**
   A. Finding the maximum depth of a binary tree
   B. Finding the minimum depth of a binary tree
   C. Finding the sum of all nodes in a binary tree
   D. Finding the lowest common ancestor of two nodes in a binary tree
   Answer: A

8. **Which of the following is an advantage of level-order traversal?**
   A. It is faster than other traversal techniques
   B. It uses less memory than other traversal techniques
   C. It can be used to find the shortest path between two nodes
   D. It can be used to sort the nodes in a binary tree
   Answer: C

9. **What is the main disadvantage of level-order traversal?**
   A. It is difficult to implement
   B. It requires more memory than other traversal techniques
   C. It is slower than other traversal techniques
   D. It does not work for binary trees with an odd number of nodes
   Answer: B

10. **What is the first node visited in level-order traversal?**
    A. The root node
    B. The left child of the root node
    C. The right child of the root node
    D. It depends on the binary tree
    Answer: A

Lec 16 - Deleting a node in BST

1. **In a BST, which node is deleted when the node to be deleted has no children?**
   a) The root node
   b) The node to be deleted
   c) The parent of the node to be deleted
   d) None of the above

Answer: b) The node to be deleted

2. **When deleting a node with one child in a BST, which child of the deleted node replaces it?**
   a) The left child
   b) The right child
   c) It depends on the node's value
   d) None of the above

Answer: c) It depends on the node's value

3. **When deleting a node with two children in a BST, which node is used to replace the deleted node**?
   a) The left child of the deleted node
   b) The right child of the deleted node
   c) The smallest node in the right subtree of the deleted node
   d) The largest node in the left subtree of the deleted node

Answer: c) The smallest node in the right subtree of the deleted node

4. **Which traversal algorithm is commonly used to delete a node in a BST?**
   a) Inorder traversal
   b) Preorder traversal
   c) Postorder traversal
   d) Level-order traversal

Answer: a) Inorder traversal

5. **In a BST, what is the time complexity of deleting a node with one child?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) It depends on the height of the tree

Answer: b) O(log n)

6. **What is the time complexity of deleting a node with two children in a BST?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) It depends on the height of the tree

Answer: d) It depends on the height of the tree

7. **What happens when a leaf node is deleted in a BST?**
   a) The node is deleted and the tree is balanced
   b) The node is deleted and the tree is left unbalanced
   c) The tree becomes a binary tree
   d) None of the above

Answer: a) The node is deleted and the tree is balanced

8. **In a self-balancing BST, what type of rotation is performed when deleting a node with one child?**
   a) Left rotation
   b) Right rotation
   c) Double rotation
   d) No rotation is performed

Answer: d) No rotation is performed

9. **When deleting a node in a BST, what is the worst-case time complexity if the tree is unbalanced?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) It depends on the size of the tree

Answer: c) O(n)

10. **In a BST, what is the minimum number of children a node can have?**
    a) 0
    b) 1
    c) 2
    d) There is no minimum number of children

Answer: a) 0

Lec 17 - Reference Variables

1. **Which of the following data types is a reference type in Java?**
   a) int
   b) double
   c) String
   d) char

Answer: c) String

2. **What does a reference variable in Java hold?**
   a) The actual value of an object
   b) The memory address of an object
   c) The size of an object
   d) The type of an object

Answer: b) The memory address of an object

3. **When an object is assigned to a reference variable, what is stored in the variable?**
   a) A copy of the object
   b) A reference to the object
   c) The value of the object
   d) None of the above

Answer: b) A reference to the object

4. **What is the difference between a reference variable and a primitive variable?**
   a) A reference variable holds the actual value of an object, while a primitive variable holds a reference to an object.
   b) A reference variable holds a reference to an object, while a primitive variable holds the actual value of a data type.
   c) A reference variable holds a value of an object, while a primitive variable holds a reference to an object.
   d) There is no difference between the two.

Answer: b) A reference variable holds a reference to an object, while a primitive variable holds the actual value of a data type.

5. **In Java, can a reference variable be null?**
   a) Yes
   b) No

Answer: a) Yes

6. **What happens when a reference variable is assigned to another reference variable?**
   a) Both variables hold a reference to the same object
   b) Both variables hold a copy of the same object
   c) The original variable is deleted
   d) None of the above

Answer: a) Both variables hold a reference to the same object

7. **What is the syntax for creating a reference variable in Java?**
   a) int x = 5;
   b) double y = 3.14;
   c) String s = "Hello";
   d) char c = 'a';

Answer: c) String s = "Hello";

8. **What happens when a reference variable is passed as a parameter to a method in Java**?
   a) The method receives a copy of the object
   b) The method receives a reference to the object
   c) The method receives the actual value of the object
   d) None of the above

Answer: b) The method receives a reference to the object

9. **What is the default value of a reference variable in Java?**
   a) null
   b) 0
   c) false
   d) None of the above

Answer: a) null

10. **Can a reference variable be used to access static methods in Java?**
    a) Yes
    b) No

Answer: a) Yes

Lec 18 - Reference Variables

1. **What is a reference variable in Java?**
   a. A variable that holds the actual value of a data type.
   b. A variable that holds the memory address of an object.
   c. A variable that holds both the value of a data type and the memory address of an object.
   Answer: b

2. **What is the default value of a reference variable in Java?**
   a. 0
   b. false
   c. null
   Answer: c

3. **Can a reference variable be reassigned to a different object in Java?**
   a. Yes
   b. No
   c. It depends on the data type of the reference variable.
   Answer: a

4. **How does Java handle passing a reference variable as a parameter to a method?**
   a. It passes the actual value of the reference variable.
   b. It passes the memory address of the object held by the reference variable.
   c. It does not allow passing reference variables as parameters to methods.
   Answer: b

5. **Can a reference variable be used to access static methods in Java?**
   a. Yes
   b. No
   c. It depends on the access modifier of the static method.
   Answer: a

6. **How does Java handle garbage collection for objects referenced by reference variables?**
   a. It automatically frees up memory allocated to objects that are no longer being referenced.
   b. It requires manual intervention to free up memory allocated to objects.
   c. It does not perform garbage collection for objects referenced by reference variables.
   Answer: a

7. **What is the difference between an instance variable and a reference variable in Java?**
   a. An instance variable holds the memory address of an object, while a reference variable is a variable declared in a class.
   b. An instance variable is a variable declared in a method, while a reference variable is a variable declared in a class.
   c. An instance variable is a variable declared in a class, while a reference variable is a variable declared in a method or block that holds a reference to an object.
   Answer: c

8. **Can a reference variable be used to access private members of a class in Java?**
   a. Yes
   b. No
   c. It depends on the access modifier of the private member.
   Answer: b

9. **How can we check if a reference variable is null in Java?**
   a. By using the null keyword.
   b. By using the equals() method.
   c. By using the == operator.
   Answer: c

10. **Can a reference variable be assigned to a primitive value in Java?**
    a. Yes
    b. No
    c. It depends on the data type of the reference variable.
    Answer: b

Lec 19 - Usage of const keyword

1. **What does the const keyword do in C++?**
   a) Declares a variable that cannot be modified
   b) Declares a variable that can only be modified in specific contexts
   c) Declares a variable that is used for debugging purposes
   Answer: a

2. **Which of the following is a benefit of using const variables?**
   a) Improved program performance
   b) Increased memory usage
   c) Reduced bugs and improved program stability
   Answer: c

3. **Can const variables be modified after they are initialized?**
   a) Yes
   b) No
   Answer: b

4. **What happens if you try to modify a const variable in C++?**
   a) The program crashes
   b) The compiler generates an error
   c) The modification is allowed
   Answer: b

5. **Which of the following types of variables is typically declared as const in C++?**
   a) Variables that are used for input/output
   b) Global variables
   c) Variables that are used as constants in the program
   Answer: c

6. **Is the const keyword required when passing a variable by reference in C++?**
   a) Yes
   b) No
   Answer: a

7. **Can a member function of a C++ class be declared as const?**
   a) Yes
   b) No
   Answer: a

8. **Which of the following is an example of a const pointer in C++?**
   a) int* const ptr;
   b) const int* ptr;
   c) const int* const ptr;
   Answer: a

9. **Can a const variable be initialized with a value at runtime in C++?**
   a) Yes
   b) No
   Answer: b

10. **Is the const keyword used in other programming languages besides C++?**
    a) Yes
    b) No
    Answer: a

Lec 20 - AVL Tree

1. **What is AVL Tree?**
   a) Binary Tree
   b) Self-balancing Binary Search Tree
   c) Hash Tree
   d) None of the above

Answer: b) Self-balancing Binary Search Tree

2. **In AVL Tree, what is the maximum difference between the height of the left and right subtrees?**
   a) 1
   b) 2
   c) 3
   d) 4

Answer: a) 1

3. **What is the time complexity of search operation in AVL Tree?**
   a) O(log n)
   b) O(n)
   c) O(n log n)
   d) O(1)

Answer: a) O(log n)

4. **In AVL Tree, what operation is performed to balance the tree?**
   a) Rotation
   b) Inversion
   c) Deletion
   d) None of the above

Answer: a) Rotation

5. **What is the worst-case time complexity of insertion operation in AVL Tree?**
   a) O(log n)
   b) O(n)
   c) O(n log n)
   d) O(1)

Answer: a) O(log n)

6. **What is the height of an AVL Tree with n nodes in the worst-case scenario?**
   a) log(n)
   b) log(n) + 1
   c) 2log(n)
   d) 2log(n) + 1

Answer: b) log(n) + 1

7. **Which of the following statements is true about AVL Tree?**
   a) AVL Tree is a balanced binary search tree
   b) AVL Tree is an unbalanced binary search tree
   c) AVL Tree is a type of heap data structure
   d) AVL Tree is a type of graph data structure

Answer: a) AVL Tree is a balanced binary search tree

8. **What is the time complexity of deletion operation in AVL Tree?**
   a) O(log n)
   b) O(n)
   c) O(n log n)
   d) O(1)

Answer: a) O(log n)

9. **Which of the following is not a balancing rule in AVL Tree?**
   a) Right-Right (RR)
   b) Left-Right (LR)
   c) Left-Left (LL)
   d) Right-Left (RL)

Answer: d) Right-Left (RL)

10. **Can AVL Tree have duplicate keys?**
    a) Yes
    b) No

Answer: b) No

Lec 21 - AVL Tree Building Example

1. **What is the first node added to the AVL Tree in the building example?**
   a) 1
   b) 2
   c) 5
   d) 8

Answer: c) 5

2. **What is the second node added to the AVL Tree in the building example?**
   a) 1
   b) 2
   c) 3
   d) 8

Answer: b) 2

3. **How many rotations are performed to maintain balance after inserting node 1 in the AVL Tree?**
   a) 0
   b) 1
   c) 2
   d) 3

Answer: b) 1

4. **What is the height of the AVL Tree after inserting node 3?**
   a) 1
   b) 2
   c) 3
   d) 4

Answer: b) 2

5. **Which rotation is performed after inserting node 6 to maintain balance in the AVL Tree?**
   a) Left rotation
   b) Right rotation
   c) Left-right rotation
   d) Right-left rotation

Answer: a) Left rotation

6. **What is the height of the AVL Tree after inserting node 9?**
   a) 3
   b) 4
   c) 5
   d) 6

Answer: b) 4

7. **What is the root node of the AVL Tree after inserting all the nodes?**
   a) 1
   b) 2
   c) 5
   d) 8

Answer: c) 5

8. **Which is the last node added to the AVL Tree in the building example?**
   a) 6
   b) 8
   c) 9
   d) 3

Answer: c) 9

9. **What is the maximum height of an AVL Tree with 7 nodes?**
   a) 2
   b) 3
   c) 4
   d) 5

Answer: b) 3

10. **How many rotations are performed in total to maintain balance while building the AVL Tree in the example?**
    a) 2
    b) 3
    c) 4
    d) 5

Answer: c) 4

Lec 22 - Cases of rotations

1. **Which of the following is not a type of rotation in a binary search tree?**
   A. Left rotation
   B. Right rotation
   C. Upward rotation
   D. Double rotation
   Answer: C

2. **When is a single left rotation used in a binary search tree?**
   A. When the imbalance occurs in the immediate left child
   B. When the imbalance occurs in the immediate right child
   C. When the imbalance occurs in the grandchild of the left child
   D. When the imbalance occurs in the grandchild of the right child
   Answer: A

3. **Which of the following is a case of double rotation in a binary search tree?**
   A. Left-Left case
   B. Left-Right case
   C. Right-Left case
   D. Right-Right case
   Answer: B

4. **In a left-right double rotation, what is the first step performed?**
   A. A single left rotation on the right child
   B. A single right rotation on the left child
   C. A double right rotation on the left child
   D. A double left rotation on the right child
   Answer: B

5. **Which of the following is not a benefit of rotations in a binary search tree?**
   A. Maintaining balance
   B. Ensuring efficient search operations
   C. Reducing the height of the tree
   D. Increasing the height of the tree
   Answer: D

6. **What is the maximum number of rotations required to balance a node in a binary search tree?**
   A. 1
   B. 2
   C. 3
   D. 4
   Answer: B

7. **When is a single right rotation used in a binary search tree?**
   A. When the imbalance occurs in the immediate left child
   B. When the imbalance occurs in the immediate right child
   C. When the imbalance occurs in the grandchild of the left child
   D. When the imbalance occurs in the grandchild of the right child
   Answer: B

8. **Which of the following is a case of single rotation in a binary search tree?**
   A. Left-Left case
   B. Left-Right case
   C. Right-Left case
   D. Right-Right case
   Answer: A

9. **In a left-left single rotation, what is the new root of the subtree?**
   A. The left child of the original root
   B. The right child of the original root
   C. The parent of the original root
   D. The original root itself
   Answer: A

10. **Which of the following is a disadvantage of using rotations in a binary search tree?**
    A. Increased tree height
    B. Reduced search efficiency
    C. Increased complexity
    D. None of the above
    Answer: D

Lec 23 - Single Right Rotation

1. **What is a single right rotation?**
   a) A rotation that balances the right child of a node.
   b) A rotation that balances the left child of a node.
   c) A rotation that balances the entire tree.
   d) A rotation that removes a node from the tree.

Answer: b) A rotation that balances the left child of a node.

2. **When is a single right rotation used?**
   a) When the imbalance occurs in the immediate left child of a node.
   b) When the imbalance occurs in the immediate right child of a node.
   c) When the tree is completely balanced.
   d) When a new node is inserted into the tree.

Answer: a) When the imbalance occurs in the immediate left child of a node.

3. **What is the purpose of a single right rotation?**
   a) To maintain the order of the nodes in the subtree.
   b) To increase the height of the left subtree.
   c) To decrease the height of the right subtree.
   d) To remove a node from the tree.

Answer: a) To maintain the order of the nodes in the subtree.

4. **What is the result of a single right rotation?**
   a) The left child becomes the new root of the subtree.
   b) The right child becomes the new root of the subtree.
   c) The subtree becomes completely balanced.
   d) A node is removed from the tree.

Answer: a) The left child becomes the new root of the subtree.

5. **What is the maximum number of rotations required to balance a node in a binary search tree?**
   a) One
   b) Two
   c) Three
   d) Four

Answer: b) Two

6. **What is the purpose of balancing a binary search tree?**
   a) To reduce the height of the tree.
   b) To increase the height of the tree.
   c) To ensure efficient search operations.
   d) To remove nodes from the tree.

Answer: c) To ensure efficient search operations.

7. **What type of rotation is used when the left child of a node has a right child and the subtree is imbalanced?**
   a) Single left rotation
   b) Single right rotation
   c) Double left rotation
   d) Double right rotation

Answer: c) Double left rotation

8. **What is the left-right case?**
   a) When the left child of a node has a right child and the subtree is imbalanced.
   b) When the right child of a node has a left child and the subtree is imbalanced.
   c) When the left child of a node has a left child and the subtree is imbalanced.
   d) When the right child of a node has a right child and the subtree is imbalanced.

Answer: a) When the left child of a node has a right child and the subtree is imbalanced.

9. **What is the right-left case?**
   a) When the right child of a node has a left child and the subtree is imbalanced.
   b) When the left child of a node has a right child and the subtree is imbalanced.
   c) When the right child of a node has a right child and the subtree is imbalanced.
   d) When the left child of a node has a left child and the subtree is imbalanced.

Answer: a) When the right child of a node has a left child and the subtree is imbalanced.

10. **What is the purpose of double rotations in a binary search tree?**
    a) To remove nodes from the tree.
    b) To maintain balance when a single rotation is not enough.
    c) To increase the height of the tree.
    d) To decrease the height of the tree.

Answer: b) To maintain balance when a single rotation is not enough.

Lec 24 - Deletion in AVL Tree

1. **What is the time complexity of deleting a node in an AVL tree?**
   a) O(n)
   b) O(log n)
   c) O(h)
   d) O(h log n)
   Answer: c) O(h)

2. **Which case is checked for rebalancing the AVL tree after a node is deleted?**
   a) Left-Left
   b) Left-Right
   c) Right-Right
   d) Right-Left
   Answer: d) Right-Left

3. **In an AVL tree, what is the maximum number of rotations needed to rebalance the tree after deleting a node?**
   a) 1
   b) 2
   c) 3
   d) 4
   Answer: b) 2

4. **Which of the following statements is true for the AVL tree?**
   a) AVL tree is a binary search tree
   b) AVL tree is a self-balancing binary search tree
   c) AVL tree is not a binary search tree
   d) AVL tree is a balanced binary search tree
   Answer: b) AVL tree is a self-balancing binary search tree

5. **What is the height of an AVL tree after deleting a node?**
   a) Remains the same
   b) Decreases by 1
   c) Increases by 1
   d) Cannot be determined
   Answer: a) Remains the same

6. **Which of the following rotations is performed when deleting a node in the Right-Right case?**
   a) Single left rotation
   b) Single right rotation
   c) Double left rotation
   d) Double right rotation
   Answer: b) Single right rotation

7. **Which of the following rotations is performed when deleting a node in the Left-Right case?**
   a) Single left rotation
   b) Single right rotation
   c) Double left rotation
   d) Double right rotation

Answer: d) Double right rotation

8. **Which of the following is a disadvantage of using AVL trees?**
   a) Faster insertion and deletion operations
   b) Slow search operation
   c) More memory space required
   d) Cannot be used to implement balanced search trees
   Answer: c) More memory space required

9. **Which of the following is an advantage of using AVL trees?**
   a) Lower time complexity for all operations
   b) Smaller tree height compared to other self-balancing trees
   c) Can handle unbalanced trees efficiently
   d) No need for tree balancing operations
   Answer: b) Smaller tree height compared to other self-balancing trees

10. **In which of the following cases is no rotation needed when deleting a node in an AVL tree?**
    a) Left-Left case
    b) Left-Right case
    c) Right-Right case
    d) Right-Left case
    Answer: c) Right-Right case

Lec 25 - Expression tree

1. **What is an expression tree?**
   a) A binary tree with nodes representing operands
   b) A binary tree with nodes representing operators
   c) A binary tree with nodes representing both operands and operators
   d) A binary tree with nodes representing numbers

Answer: c

2. **What is the purpose of an expression tree?**
   a) To represent a mathematical expression
   b) To store data in a tree structure
   c) To sort data in a binary tree
   d) To perform search operations on data in a binary tree

Answer: a

3. **Which traversal of an expression tree is used to evaluate the expression?**
   a) Preorder
   b) Inorder
   c) Postorder
   d) Level order

Answer: c

4. **What is the time complexity of evaluating an expression tree?**
   a) O(n)
   b) O(log n)
   c) O(n^2)
   d) O(2^n)

Answer: a

5. **How is an expression tree created from an infix expression?**
   a) Using the preorder traversal
   b) Using the inorder traversal
   c) Using the postorder traversal
   d) Using a stack

Answer: d

6. **What is the maximum number of children a node in an expression tree can have?**
   a) 0
   b) 1
   c) 2
   d) 3

Answer: c

7. **Which of the following operations can be performed on an expression tree?**
   a) Insertion of a node
   b) Deletion of a node
   c) Rotation of a node
   d) All of the above

Answer: d

8. **What is the purpose of a leaf node in an expression tree?**
   a) To represent an operator
   b) To represent an operand
   c) To represent a binary operation
   d) To represent a unary operation

Answer: b

9. **Can an expression tree have duplicate nodes?**
   a) Yes
   b) No

Answer: b

10. **What is the advantage of using an expression tree over a postfix expression?**
    a) Faster evaluation
    b) Easier to read
    c) Takes less space
    d) All of the above

Answer: d

Lec 26 - Hoffman Encoding

1. **Which of the following is a lossless data compression algorithm?**
   a) Huffman encoding
   b) Arithmetic encoding
   c) Run-length encoding
   d) All of the above

Answer: d) All of the above

2. **In Huffman encoding, the symbols to be encoded are represented as what?**
   a) A binary tree
   b) A prefix code
   c) A suffix code
   d) None of the above

Answer: a) A binary tree

3. **What is the goal of Huffman encoding?**
   a) To compress data by representing frequently occurring symbols in a compressed form
   b) To convert data from analog to digital form
   c) To encrypt data for secure transmission
   d) None of the above

Answer: a) To compress data by representing frequently occurring symbols in a compressed form

4. **How is the Huffman tree built?**
   a) By merging the two most frequent symbols at each step
   b) By merging the two least frequent symbols at each step
   c) By randomly selecting symbols to be included
   d) None of the above

Answer: b) By merging the two least frequent symbols at each step

5. **Which of the following is NOT a property of the Huffman code?**
   a) Prefix-free
   b) Uniquely decodable
   c) Provides a compact representation of the original data
   d) Lossy compression

Answer: d) Lossy compression

6. **What is a prefix code in Huffman encoding?**
   a) A code in which no codeword is a prefix of any other codeword
   b) A code in which each symbol is represented by the same number of bits
   c) A code in which the codewords are sorted in order of frequency
   d) None of the above

Answer: a) A code in which no codeword is a prefix of any other codeword

7. **Which of the following is a disadvantage of Huffman encoding?**
   a) It requires the entire input to be available at once
   b) It is slow to encode and decode
   c) It cannot be used with binary data
   d) None of the above

Answer: a) It requires the entire input to be available at once

8. **What is the time complexity of building a Huffman tree?**
   a) O(n)
   b) O(n log n)
   c) O(n^2)
   d) None of the above

Answer: b) O(n log n)

9. **What is the space complexity of building a Huffman tree?**
   a) O(n)
   b) O(log n)
   c) O(n log n)
   d) None of the above

Answer: a) O(n)

10. **Which of the following is an application of Huffman encoding?**
    a) Lossy audio compression
    b) Lossless image compression
    c) Data encryption
    d) None of the above

Answer: b) Lossless image compression

Lec 27 - Properties of Binary Tree

1. **Which of the following statements is true about a binary tree?**
   A. Each node has exactly two children
   B. Each node has at most two children
   C. Each node has at least two children
   D. Each node has exactly one child

Answer: B

2. **In a binary tree, a node is said to be a leaf node if:**
   A. It has no children
   B. It has exactly one child
   C. It has at least one child
   D. It has two children

Answer: A

3. **A binary tree is said to be a complete binary tree if:**
   A. All the nodes in the tree have the same value
   B. Each node has at most two children
   C. Each level of the tree is completely filled
   D. The tree is balanced

Answer: C

4. **Which of the following traversal methods visits the left subtree, then the root, and finally the right subtree?**
   A. Preorder
   B. Inorder
   C. Postorder
   D. Level order

Answer: B

5. **A binary tree is said to be balanced if:**
   A. All the nodes have the same value
   B. Each node has at most two children
   C. The height of the left and right subtrees of any node differ by at most 1
   D. The tree is complete

Answer: C

6. **In a binary tree, the maximum number of nodes at level k is:**
   A. 2^k
   B. k^2
   C. 2k
   D. 2^(k-1)

Answer: A

7. **The number of edges in a full binary tree with n nodes is:**
   A. n-1
   B. n
   C. 2n-1
   D. 2n

Answer: C

8. **Which of the following statements is true about a binary search tree?**
   A. Each node has at most two children
   B. The left subtree of a node contains only nodes with values less than the node's value
   C. The right subtree of a node contains only nodes with values greater than the node's value
   D. All of the above

Answer: D

9. **A binary tree in which every non-leaf node has non-empty left and right subtrees is called a:**
   A. Full binary tree
   B. Complete binary tree
   C. Balanced binary tree
   D. None of the above

Answer: A

10. **In a binary tree, the height is defined as:**
    A. The number of nodes in the tree
    B. The number of edges from the root to the farthest leaf node
    C. The number of levels in the tree
    D. The number of subtrees in the tree

Answer: B

Lec 28 - Inorder traversal in threaded trees

1. **In threaded binary tree, a node that has no left child and whose left pointer points to the _____.**
   a) in-order predecessor
   b) in-order successor
   c) null
   d) none of the mentioned
   Answer: a

2. **What is a threaded binary tree?**
   a) A binary tree in which each node can have any number of children
   b) A binary tree in which all the left pointers point to inorder predecessors and right pointers point to inorder successors.
   c) A binary tree in which each node can have at most 2 children
   d) A binary tree in which all the leaf nodes have a level of 0.
   Answer: b

3. **What is the time complexity for finding the inorder successor in a threaded binary tree?**
   a) O(1)
   b) O(n)
   c) O(log n)
   d) O(n^2)
   Answer: a

4. **In which traversal, the nodes are visited in increasing order of their values?**
   a) Inorder Traversal
   b) Preorder Traversal
   c) Postorder Traversal
   d) Level order Traversal
   Answer: a

5. **In threaded binary trees, the right pointer of a node points to its _____.**
   a) Predecessor
   b) Successor
   c) Ancestor
   d) Descendant
   Answer: b

6. **Which of the following is not true for threaded binary trees?**
   a) In-order traversal can be performed in O(n) time complexity
   b) They save storage space
   c) They are more efficient than normal binary trees for finding in-order predecessors and successors.
   d) They allow for easy deletion of a node
   Answer: d

7. **Which of the following is a disadvantage of threaded binary trees?**
   a) They take up more space than normal binary trees
   b) They are less efficient than normal binary trees for finding in-order predecessors and successors.
   c) They make deletion of a node difficult.

d) They require extra memory space to store the thread pointers.
Answer: d

8. **What is the main advantage of using threaded binary trees?**
   a) They are easier to implement than normal binary trees
   b) They allow for efficient finding of in-order predecessors and successors
   c) They have a shorter height than normal binary trees
   d) They can store more data than normal binary trees
   Answer: b

9. **Which of the following is not a type of threaded binary tree?**
   a) Single threaded binary tree
   b) Double threaded binary tree
   c) Circular threaded binary tree
   d) Quadruple threaded binary tree
   Answer: d

10. **Which of the following is not true for threaded binary trees?**
    a) They are used for storing large amounts of data
    b) They allow for efficient traversal of the tree
    c) They can be used for faster searching of data
    d) They have a shorter height than normal binary trees.
    Answer: a

Lec 29 - Complete Binary Tree

1. **A binary tree is said to be complete if:**
   a) All nodes have two children
   b) All levels are completely filled except possibly the last level
   c) All nodes have at most two children
   d) None of the above

Answer: b) All levels are completely filled except possibly the last level

2. **What is the maximum number of nodes a complete binary tree of height h can have?**
   a) 2^h+1
   b) 2^h-1
   c) h^2
   d) None of the above

Answer: b) 2^h-1

3. **What is the minimum number of nodes a complete binary tree of height h can have?**
   a) 2^(h-1)
   b) 2^(h-1)-1
   c) h^2-1
   d) None of the above

Answer: a) 2^(h-1)

4. **A complete binary tree of height h has _____ leaf nodes.**
   a) 2^h-1
   b) 2^(h-1)
   c) 2^(h-1)+1
   d) None of the above

Answer: b) 2^(h-1)

5. **What is the height of a complete binary tree with 15 nodes?**
   a) 3
   b) 4
   c) 5
   d) None of the above

Answer: b) 4

6. **A complete binary tree can be efficiently stored in an array using:**
   a) Inorder traversal
   b) Preorder traversal
   c) Postorder traversal
   d) Level order traversal

Answer: d) Level order traversal

7. **The number of internal nodes in a complete binary tree of height h is:**
   a) 2^h
   b) 2^h-1
   c) 2^(h+1)-1
   d) None of the above

Answer: b) 2^h-1

8. **What is the parent of the node at index i in an array representation of a complete binary tree?**
   a) i-1
   b) i/2
   c) 2*i
   d) None of the above

Answer: b) i/2

9. **A complete binary tree of n nodes has its root at index:**
   a) 0
   b) 1
   c) n-1
   d) n

Answer: a) 0

10. **Which of the following is NOT true about a complete binary tree?**
    a) It can have a maximum of 2^h-1 nodes
    b) It can have a minimum of 2^(h-1) nodes
    c) Its last level can have any number of nodes
    d) All levels except possibly the last level are completely filled

Answer: c) Its last level can have any number of nodes

Lec 30 - Inserting into a Min-Heap

1. **In a min-heap, the root node always contains the _____ element.**
   a) Maximum
   b) Minimum
   c) Median
   d) Random

Answer: b) Minimum

2. **The worst-case time complexity for inserting an element in a min-heap is:**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) O(n log n)

Answer: b) O(log n)

3. **Which property of a min-heap ensures that the root node always contains the minimum element?**
   a) Complete binary tree property
   b) Heap order property
   c) Both (a) and (b)
   d) None of the above

Answer: b) Heap order property

4. **To insert an element in a min-heap, we always add it to the:**
   a) Leftmost position at the deepest level
   b) Rightmost position at the deepest level
   c) Leftmost position at the second deepest level
   d) Rightmost position at the second deepest level

Answer: a) Leftmost position at the deepest level

5. **If we insert the elements 8, 5, 3, 9, 1, 7, 6, 2 in a min-heap, what will be the root node?**
   a) 1
   b) 2
   c) 3
   d) 5

Answer: a) 1

6. **The height of a min-heap with n elements is:**
   a) log n
   b) n/2
   c) n-1
   d) n

Answer: a) log n

7. **Which of the following operations is NOT supported by a min-heap?**
   a) Insertion
   b) Deletion
   c) Search
   d) All of the above

Answer: c) Search

8. **To maintain the heap order property after inserting an element, we perform:**
   a) Up-heap bubbling
   b) Down-heap bubbling
   c) Both (a) and (b)
   d) None of the above

Answer: a) Up-heap bubbling

9. **If we insert an element in a min-heap, the new element will always be a**:
   a) Leaf node
   b) Parent node
   c) Child node
   d) Sibling node

Answer: a) Leaf node

10. **The time complexity of building a min-heap from an array of n elements is:**
    a) O(1)
    b) O(n)
    c) O(n log n)
    d) O(n^2)

Answer: b) O(n)

Lec 31 - BuildHeap

1. **What is the time complexity of BuildHeap algorithm?**
   a) O(n log n)
   b) O(n^2)
   c) O(n)
   d) O(log n)

Answer: c) O(n)

2. **Which data structure is created by BuildHeap algorithm?**
   a) Array
   b) Linked List
   c) Tree
   d) Heap

Answer: d) Heap

3. **What is the maximum number of swaps required in BuildHeap algorithm?**
   a) n-1
   b) n
   c) n/2
   d) log n

Answer: b) n

4. **Which sorting algorithm uses BuildHeap internally?**
   a) Insertion Sort
   b) Merge Sort
   c) Quick Sort
   d) Heap Sort

Answer: d) Heap Sort

5. **What is the worst-case time complexity of HeapSort?**
   a) O(n log n)
   b) O(n^2)
   c) O(n)
   d) O(log n)

Answer: a) O(n log n)

6. **Which property does a heap satisfy?**
   a) All nodes are greater than their parent nodes
   b) All nodes are less than their parent nodes
   c) All nodes are equal to their parent nodes
   d) None of the above

Answer: a) All nodes are greater than their parent nodes

7. **What is the index of the last non-leaf node in a binary heap?**
   a) (n-1)/2
   b) (n-2)/2
   c) n/2
   d) n-2

Answer: b) (n-2)/2

8. **Which operation is used to remove the root element from a heap?**
   a) Delete
   b) ExtractMin/ExtractMax
   c) Pop
   d) Remove

Answer: b) ExtractMin/ExtractMax

9. **Which data structure is best suited for implementing a priority queue?**
   a) Stack
   b) Queue
   c) Heap
   d) Linked List

Answer: c) Heap

10. **What is the worst-case time complexity of inserting an element in a heap?**
    a) O(log n)
    b) O(n)
    c) O(n log n)
    d) O(1)

Answer: a) O(log n)

Lec 32 - perculateDown Method

1. **What is the purpose of the percolateDown method in a heap data structure?**
   A. To insert an element into the heap.
   B. To maintain the heap property after removing the root element.
   C. To sort the elements in the heap.
   D. None of the above.

Answer: B

2. **What is the time complexity of the percolateDown method?**
   A. O(n)
   B. O(log n)
   C. O(n log n)
   D. O(1)

Answer: B

3. **Which element is swapped with the root element in the percolateDown method?**
   A. The smallest child element
   B. The largest child element
   C. The first element in the heap
   D. None of the above

Answer: B

4. **What happens if the root element has no children in the percolateDown method?**
   A. The root element is removed from the heap.
   B. The heap is left unchanged.
   C. An error is thrown.
   D. None of the above.

Answer: B

5. **Is the percolateDown method used in HeapSort algorithm?**
   A. Yes
   B. No

Answer: A

6. **Which type of heap data structure is percolateDown method used for?**
   A. Max heap
   B. Min heap
   C. Both
   D. Neither

Answer: C

7. **Does the percolateDown method modify the size of the heap data structure?**
   A. Yes

B. No

Answer: A

8. **How many elements are swapped at most in the percolateDown method?**
   A. One
   B. Two
   C. Three
   D. Four

Answer: B

9. **Is the percolateDown method a recursive algorithm?**
   A. Yes
   B. No

Answer: A

10. **What is the worst-case time complexity of the percolateDown method?**
    A. O(n)
    B. O(log n)
    C. O(n log n)
    D. O(1)

Answer: B

Lec 33 - Priority Queue Using Heap

1. **What is a priority queue using a heap?**
   A) A queue where elements are arranged in the order they are inserted
   B) A queue where elements are arranged in ascending order
   C) A queue where elements are arranged based on their priority
   D) A queue where elements are arranged in descending order
   Answer: C

2. **Which operation(s) can be performed on a priority queue?**
   A) Insertion
   B) Deletion
   C) Retrieval of the highest priority element
   D) All of the above
   Answer: D

3. **What is the time complexity of insertion in a priority queue using a heap?**
   A) O(1)
   B) O(log n)
   C) O(n)
   D) O(n^2)
   Answer: B

4. **What is the time complexity of retrieval of the highest priority element in a priority queue using a heap?**
   A) O(1)
   B) O(log n)
   C) O(n)
   D) O(n^2)
   Answer: A

5. **Which data structure is used to implement a priority queue using a heap?**
   A) Array
   B) Linked list
   C) Stack
   D) Queue
   Answer: A

6. **What is the property of a heap that ensures the highest priority element is always at the top?**
   A) Heap size
   B) Heap capacity
   C) Heap order
   D) Heap property
   Answer: D

7. **Which type of heap is used to implement a priority queue?**
   A) Max heap
   B) Min heap
   C) Both A and B
   D) Neither A nor B
   Answer: A

8. **What happens when a new element is inserted into a priority queue using a heap?**
   A) The element is added to the end of the heap
   B) The element is added to the beginning of the heap
   C) The element is added to the correct position based on its priority
   D) None of the above
   Answer: C

9. **What happens when the highest priority element is removed from a priority queue using a heap?**
   A) The last element is removed
   B) The first element is removed
   C) The element in the correct position is removed
   D) None of the above
   Answer: C

10. **Which of the following statements is true about a priority queue using a heap?**
    A) The elements are arranged in ascending order
    B) The time complexity of insertion is O(n)
    C) The highest priority element is always at the top
    D) All elements have the same priority
    Answer: C

Lec 34 - Equivalence Relations

1. **Which of the following is not a property of an equivalence relation?**
   a. Reflexivity
   b. Symmetry
   c. Transitivity
   d. Antisymmetry
   Answer: d. Antisymmetry

2. **Which of the following is an example of an equivalence relation?**
   a. Greater than
   b. Less than
   c. Equality
   d. Addition
   Answer: c. Equality

3. **An equivalence class is a set of elements that:**
   a. Have the same value
   b. Are not related to each other
   c. Have the same property
   d. Have different properties
   Answer: a. Have the same value

4. **Which of the following is an example of a relation that is not an equivalence relation?**
   a. Greater than or equal to
   b. Less than or equal to
   c. Not equal to
   d. None of the above
   Answer: c. Not equal to

5. **If xRy and yRz, then xRz is an example of which property of an equivalence relation?**
   a. Reflexivity
   b. Symmetry
   c. Transitivity
   d. None of the above
   Answer: c. Transitivity

6. **Which of the following is an example of a partition of a set?**
   a. {1, 2, 3}, {4, 5}, {6, 7, 8}
   b. {1, 3, 5}, {2, 4, 6}
   c. {a, b, c}, {d, e}
   d. All of the above
   Answer: a. {1, 2, 3}, {4, 5}, {6, 7, 8}

7. **An equivalence relation can be defined on which of the following sets?**
   a. Integers
   b. Rational numbers
   c. Real numbers
   d. All of the above
   Answer: d. All of the above

8. **Which of the following is a common use of equivalence relations in computer science?**
   a. Database design
   b. Sorting algorithms
   c. Graph theory
   d. Cryptography
   Answer: a. Database design

9. **Which of the following is an example of a non-trivial equivalence relation?**
   a. Equality
   b. Greater than
   c. Less than
   d. Congruence modulo n
   Answer: d. Congruence modulo n

10. **Which of the following is an example of an equivalence relation on a set of colors?**
    a. Lighter than
    b. Darker than
    c. Same hue
    d. None of the above
    Answer: c. Same hue

Lec 35 - Dynamic Equivalence Problem

1. **Which of the following best describes the dynamic equivalence problem?**
   a) The problem of finding the minimum number of equivalence classes for a given set of elements.
   b) The problem of efficiently maintaining equivalence relations under dynamic changes to a set of elements.
   c) The problem of finding the maximum number of equivalence classes for a given set of elements.
   d) The problem of determining the transitive closure of a given relation.

Answer: b

2. **Which data structure is commonly used to solve the dynamic equivalence problem?**
   a) Arrays
   b) Linked lists
   c) Binary search trees
   d) Disjoint-set data structures

Answer: d

3. **What is the time complexity of finding the equivalence class of an element using a disjoint-set data structure?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) O(n log n)

Answer: b

4. **Which operation is used to combine two equivalence classes into a single equivalence class in a disjoint-set data structure?**
   a) Make set
   b) Find set
   c) Union
   d) Intersection

Answer: c

5. **Which of the following is not a step in the path compression technique used in disjoint-set data structures?**
   a) Traverse the path from the root to the node.
   b) Set the parent of each node in the path to the root.
   c) Set the rank of each node in the path to zero.
   d) Update the rank of the root node.

Answer: c

6. **Which of the following is an advantage of using a rank-based union technique in disjoint-set data structures?**

a) Reduced time complexity of the find operation
b) Reduced time complexity of the union operation
c) Reduced memory usage
d) Improved scalability

Answer: b

7. **What is the time complexity of the union operation using a rank-based union technique in disjoint-set data structures?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) O(n log n)

Answer: b

8. **Which of the following is not a modification to the standard disjoint-set data structure that can improve its performance?**
   a) Path compression
   b) Rank-based union
   c) Weighted union
   d) Node reordering

Answer: d

9. **Which of the following statements is true about the dynamic equivalence problem?**
   a) It can be solved efficiently using brute force algorithms.
   b) It can only be solved using advanced data structures and algorithms.
   c) It is a well-defined problem that has a unique solution.
   d) It has no practical applications in computer science.

Answer: b

10. **Which of the following areas of computer science does not involve solving the dynamic equivalence problem?**
    a) Databases
    b) Information retrieval
    c) Natural language processing
    d) Computer graphics

Answer: d

Lec 36 - Running Time Analysis

1. **What is the time complexity of a linear search algorithm?**
   a) O(n)
   b) O(log n)
   c) O(n^2)
   d) O(1)
   Answer: a) O(n)

2. **Which of the following is not an asymptotic notation used for running time analysis?**
   a) Big O notation
   b) Theta notation
   c) Little O notation
   d) Epsilon notation
   Answer: d) Epsilon notation

3. **What is the time complexity of a binary search algorithm?**
   a) O(n)
   b) O(log n)
   c) O(n^2)
   d) O(1)
   Answer: b) O(log n)

4. **Which of the following is a constant time complexity algorithm?**
   a) Bubble sort
   b) Insertion sort
   c) Quick sort
   d) Counting sort
   Answer: d) Counting sort

5. **Which of the following is an example of an exponential time complexity algorithm?**
   a) Merge sort
   b) Quick sort
   c) Bubble sort
   d) Traveling salesman problem
   Answer: d) Traveling salesman problem

6. **Which of the following is not a factor that can affect the running time of an algorithm?**
   a) Input size
   b) Memory usage
   c) Hardware configuration
   d) Implementation details
   Answer: b) Memory usage

7. **What is the time complexity of a worst-case scenario for a sorting algorithm?**
   a) O(n)
   b) O(log n)
   c) O(n^2)
   d) O(1)
   Answer: c) O(n^2)

8. **What is the time complexity of a best-case scenario for a sorting algorithm**?
   a) O(n)
   b) O(log n)
   c) O(n^2)
   d) O(1)
   Answer: a) O(n)

9. **Which of the following is an example of a logarithmic time complexity algorithm?**
   a) Merge sort
   b) Quick sort
   c) Binary search
   d) Bubble sort
   Answer: c) Binary search

10. **Which of the following is an example of a quadratic time complexity algorithm?**
    a) Merge sort
    b) Quick sort
    c) Insertion sort
    d) Heap sort
    Answer: c) Insertion sort

Lec 37 - Review

1. **What is the purpose of a review?**
   A. To advertise a product
   B. To evaluate or assess something
   C. To make a sale
   D. To provide customer support
   Answer: B

2. **Which of the following is NOT a channel for expressing reviews?**
   A. Social media
   B. Online platforms
   C. Word of mouth
   D. Direct mail
   Answer: D

3. **What is the tone of a positive review?**
   A. Critical
   B. Negative
   C. Neutral
   D. Praising
   Answer: D

4. **Why are reviews important for consumers?**
   A. To increase the price of products
   B. To make informed decisions about products or services
   C. To deceive customers
   D. To limit product availability
   Answer: B

5. **What is the purpose of negative reviews?**
   A. To promote a product
   B. To evaluate or assess something
   C. To provide customer support
   D. To warn others about potential issues
   Answer: D

6. **Which of the following is an example of a review platform?**
   A. Amazon
   B. Twitter
   C. LinkedIn
   D. YouTube
   Answer: A

7. **What is the importance of customer feedback in reviews?**
   A. To make sales
   B. To improve products or services
   C. To deceive customers
   D. To increase the price of products
   Answer: B

8. **What is the tone of a neutral review?**
   A. Critical
   B. Negative
   C. Neutral
   D. Praising
   Answer: C

9. **Why are reviews important for businesses?**
   A. To limit customer feedback
   B. To decrease product availability
   C. To understand customer feedback and improve offerings
   D. To increase product price
   Answer: C

10. **What is the purpose of a review aggregator?**
    A. To increase the price of products
    B. To deceive customers
    C. To provide customer support
    D. To collect and summarize reviews from multiple sources
    Answer: D

Lec 38 - Table and Dictionaries

1. **Which data structure is best suited for storing large amounts of data that can be easily searched and sorted?**
   A) Arrays
   B) Tables
   C) Linked lists
   D) Trees

Answer: B) Tables

2. **Which data structure is best suited for storing configuration data or creating lookup tables?**
   A) Arrays
   B) Tables
   C) Linked lists
   D) Trees

Answer: B) Tables

3. **Which data structure is used to store key-value pairs?**
   A) Arrays
   B) Tables
   C) Linked lists
   D) Dictionaries

Answer: D) Dictionaries

4. **In a dictionary, what does the key represent?**
   A) The value being stored
   B) The position in memory where the value is stored
   C) The identifier used to retrieve the value
   D) The data type of the value being stored

Answer: C) The identifier used to retrieve the value

5. **Which of the following is not a common operation performed on a dictionary?**
   A) Adding a new key-value pair
   B) Removing a key-value pair
   C) Sorting the dictionary
   D) Updating the value of an existing key-value pair

Answer: C) Sorting the dictionary

6. **Which of the following is true about tables?**
   A) They can only store numeric data
   B) They are similar to linked lists
   C) They are often used to store large amounts of data
   D) They cannot be searched or sorted

Answer: C) They are often used to store large amounts of data

7. **Which of the following data structures is most similar to a table?**
   A) Arrays
   B) Stacks
   C) Queues
   D) Linked lists

Answer: A) Arrays

8. **Which of the following is an advantage of using a dictionary over a table?**
   A) Dictionaries can store more data than tables
   B) Dictionaries are easier to search and sort than tables
   C) Dictionaries are more efficient for storing numerical data
   D) Dictionaries are more flexible for storing data in key-value pairs

Answer: D) Dictionaries are more flexible for storing data in key-value pairs

9. **Which of the following is an example of a key-value pair in a dictionary?**
   A) "John", "Doe"
   B) "apple", 3.99
   C) "red", "blue", "green"
   D) 123, "456"

Answer: B) "apple", 3.99

10. **Which data structure would be best suited for storing a list of items in the order they were added?**
    A) Arrays
    B) Tables
    C) Linked lists
    D) Dictionaries

Answer: C) Linked lists

Lec 39 - Searching an Array: Binary Search

1. **What is the time complexity of binary search algorithm?**
   a) O(1)
   b) O(n)
   c) O(log n)
   d) O(n^2)
   Answer: c) O(log n)

2. **In which type of array is binary search the most efficient?**
   a) Sorted array
   b) Unsorted array
   c) Randomly sorted array
   d) None of the above
   Answer: a) Sorted array

3. **Binary search algorithm can be used for:**
   a) Array
   b) Linked list
   c) Both A and B
   d) None of the above
   Answer: a) Array

4. **Binary search algorithm can be applied to:**
   a) Characters
   b) Integers
   c) Floats
   d) All of the above
   Answer: d) All of the above

5. **Which of the following is not a step in binary search algorithm?**
   a) Check if the middle element is equal to the target element
   b) If the target element is greater than the middle element, search the left half of the array
   c) If the target element is less than the middle element, search the right half of the array
   d) Return the index of the target element
   Answer: d) Return the index of the target element

6. **What is the worst-case time complexity of binary search algorithm?**
   a) O(1)
   b) O(n)
   c) O(log n)
   d) O(n^2)
   Answer: c) O(log n)

7. **Which of the following is not a requirement for binary search algorithm to work?**
   a) The array must be sorted
   b) The array must be in ascending order
   c) The array must be in descending order
   d) The array must be homogeneous
   Answer: c) The array must be in descending order

8. **What is the middle element in an array of size 10?**
   a) 4
   b) 5
   c) 9
   d) 10
   Answer: b) 5

9. **How many elements are left in the array after the first iteration of binary search on an array of size 16?**
   a) 8
   b) 4
   c) 2
   d) 1
   Answer: a) 8

10. **What is the index of the target element in the array [1, 3, 5, 7, 9] when using binary search to find 7?**
    a) 2
    b) 3
    c) 4
    d) 5
    Answer: b) 3

Lec 40 - Skip List

1. **Which of the following data structures is a probabilistic data structure?**
   A) Binary search tree
   B) AVL tree
   C) Skip list
   D) Red-black tree
   Answer: C) Skip list

2. **In a skip list, what is the maximum number of levels that a node can have?**
   A) 1
   B) 2
   C) log n
   D) Unlimited
   Answer: D) Unlimited

3. **What is the time complexity of searching for an element in a skip list?**
   A) O(n)
   B) O(log n)
   C) O(n log n)
   D) O(1)
   Answer: B) O(log n)

4. **Which of the following operations cannot be performed on a skip list?**
   A) Insertion
   B) Deletion
   C) Searching
   D) Sorting
   Answer: D) Sorting

5. **Which of the following is the main advantage of using skip lists over balanced trees?**
   A) Space efficiency
   B) Time efficiency
   C) Ease of implementation
   D) None of the above
   Answer: C) Ease of implementation

6. **Which of the following is a disadvantage of using skip lists?**
   A) High space complexity
   B) High time complexity
   C) Limited applicability
   D) None of the above
   Answer: A) High space complexity

7. **In a skip list, what is the probability of a node having k+1 levels, given that it has k levels?**
   A) 1/2
   B) 1/4
   C) 1/8
   D) 1/16
   Answer: B) 1/4

8. **What is the worst-case time complexity of insertion in a skip list?**
   A) O(n)
   B) O(log n)
   C) O(n log n)
   D) O(1)
   Answer: A) O(n)

9. **In a skip list, what is the maximum number of nodes that can be present in a level i, given that there are n total nodes in the skip list?**
   A) n
   B) n/2
   C) n/log n
   D) log n
   Answer: B) n/2

10. **Which of the following is a disadvantage of using skip lists over hash tables?**
    A) Lower space complexity
    B) Higher time complexity
    C) Lack of support for efficient range queries
    D) None of the above
    Answer: C) Lack of support for efficient range queries

Lec 41 - Review

1. **What is the purpose of a review?**
   A. To advertise a product
   B. To evaluate or assess something
   C. To make a sale
   D. To provide customer support
   Answer: B

2. **Which of the following is NOT a channel for expressing reviews?**
   A. Social media
   B. Online platforms
   C. Word of mouth
   D. Direct mail
   Answer: D

3. **What is the tone of a positive review?**
   A. Critical
   B. Negative
   C. Neutral
   D. Praising
   Answer: D

4. **Why are reviews important for consumers?**
   A. To increase the price of products
   B. To make informed decisions about products or services
   C. To deceive customers
   D. To limit product availability
   Answer: B

5. **What is the purpose of negative reviews?**
   A. To promote a product
   B. To evaluate or assess something
   C. To provide customer support
   D. To warn others about potential issues
   Answer: D

6. **Which of the following is an example of a review platform?**
   A. Amazon
   B. Twitter
   C. LinkedIn
   D. YouTube
   Answer: A

7. **What is the importance of customer feedback in reviews?**
   A. To make sales
   B. To improve products or services
   C. To deceive customers
   D. To increase the price of products
   Answer: B

8. **What is the tone of a neutral review?**
   A. Critical
   B. Negative
   C. Neutral
   D. Praising
   Answer: C

9. **Why are reviews important for businesses?**
   A. To limit customer feedback
   B. To decrease product availability
   C. To understand customer feedback and improve offerings
   D. To increase product price
   Answer: C

10. **What is the purpose of a review aggregator?**
    A. To increase the price of products
    B. To deceive customers
    C. To provide customer support
    D. To collect and summarize reviews from multiple sources
    Answer: D

Lec 42 - Collision

1. **What is collision in computer science?**
   A. A situation where a program crashes
   B. A situation where two or more data items end up at the same memory location
   C. A situation where a program encounters a syntax error
   D. A situation where a program encounters a logical error
   Answer: B

2. **Which of the following data structures can experience collisions?**
   A. Linked lists
   B. Arrays
   C. Hash tables
   D. Stacks
   Answer: C

3. **What is the impact of collisions on data structure performance?**
   A. Faster access times
   B. Slower access times
   C. Increased data security
   D. Decreased memory consumption
   Answer: B

4. **What is chaining in collision handling?**
   A. Resolving a collision by allocating new memory
   B. Resolving a collision by reorganizing the data structure
   C. Resolving a collision by deleting the collided data item
   D. Resolving a collision by linking the collided data items together
   Answer: D

5. **Which of the following is a disadvantage of chaining?**
   A. It requires less memory
   B. It can result in longer access times
   C. It can result in data loss
   D. It requires more processing power
   Answer: B

6. **Which of the following is a disadvantage of open addressing?**
   A. It requires more memory
   B. It can result in longer access times
   C. It can result in data loss
   D. It requires more processing power
   Answer: A

7. **What is linear probing in open addressing?**
   A. Resolving a collision by rehashing the key
   B. Resolving a collision by allocating new memory
   C. Resolving a collision by searching sequentially for an empty slot
   D. Resolving a collision by randomly selecting a new memory location
   Answer: C

8. **What is quadratic probing in open addressing?**
   A. Resolving a collision by rehashing the key
   B. Resolving a collision by allocating new memory
   C. Resolving a collision by searching sequentially for an empty slot
   D. Resolving a collision by incrementing the probe step by a quadratic function of the previous step
   Answer: D

9. **Which of the following is an example of a hash function?**
   A. Sorting algorithm
   B. Linear search
   C. Bubble sort
   D. MD5
   Answer: D

10. **What is the purpose of a hash function in collision handling?**
    A. To reduce the number of collisions
    B. To increase the number of collisions
    C. To increase the memory consumption
    D. To decrease the access time
    Answer: A

Lec 43 - Hashing Animation

1. **What is the purpose of hashing animation?**
   A) To visualize the working of a hash table
   B) To test the efficiency of a hash function
   C) To optimize the performance of a hash table
   D) To compare different sorting algorithms
   Answer: A

2. **Which of the following is a common technique for handling collisions in hash tables?**
   A) Sorting
   B) Merging
   C) Chaining
   D) Selection
   Answer: C

3. **In hashing animation, what does a slot in a hash table represent?**
   A) A key-value pair
   B) A hash code
   C) A collision
   D) A search operation
   Answer: A

4. **How does a hash function map keys to their corresponding slots in a hash table?**
   A) By performing a mathematical operation on the key
   B) By iterating over each slot in the table
   C) By using a binary search algorithm
   D) By comparing the key to a pre-defined list of values
   Answer: A

5. **Which of the following is an advantage of using hashing over other data structures like arrays or linked lists?**
   A) Constant time complexity for all operations
   B) Lower space complexity
   C) More flexible data organization
   D) Greater accuracy in data retrieval
   Answer: A

6. **What is the role of a load factor in a hash table?**
   A) To control the number of collisions
   B) To determine the number of slots in the table
   C) To improve the performance of the hash function
   D) To ensure the hash table is sorted
   Answer: A

7. **Which of the following is a disadvantage of using chaining to handle collisions in a hash table?**
   A) Increased space complexity
   B) Decreased search efficiency
   C) More complex implementation
   D) Lower load factor
   Answer: A

8. **What is open addressing in hash tables?**
   A) A technique for handling collisions by storing values in linked lists
   B) A technique for handling collisions by searching for the next available slot in the table
   C) A technique for handling collisions by rehashing the key
   D) A technique for handling collisions by using a binary search algorithm
   Answer: B

9. **In a hash table, what is the worst-case time complexity for a search operation?**
   A) O(1)
   B) O(log n)
   C) O(n)
   D) It depends on the specific hash function
   Answer: C

10. **How does the performance of a hash table change as the load factor increases?**
    A) It becomes faster
    B) It becomes slower
    C) It remains constant
    D) It depends on the specific hash function
    Answer: B

Lec 44 - Selection Sort

1. **What is the time complexity of selection sort?**
   a) O(n)
   b) O(n log n)
   c) O(n^2)
   d) O(2^n)
   Answer: c) O(n^2)

2. **Which of the following is true about selection sort?**
   a) It is an in-place sorting algorithm
   b) It is a stable sorting algorithm
   c) It is a divide-and-conquer sorting algorithm
   d) It is a comparison-based sorting algorithm
   Answer: d) It is a comparison-based sorting algorithm

3. **Which of the following is the best case time complexity of selection sort?**
   a) O(n)
   b) O(n log n)
   c) O(n^2)
   d) O(2^n)
   Answer: c) O(n^2)

4. **Which of the following data structures is commonly used to implement selection sort?**
   a) Array
   b) Linked List
   c) Stack
   d) Queue
   Answer: a) Array

5. **Which of the following is the space complexity of selection sort?**
   a) O(n)
   b) O(log n)
   c) O(1)
   d) O(n log n)
   Answer: c) O(1)

6. **Which of the following is the first step in selection sort?**
   a) Compare the first two elements
   b) Find the smallest element in the array
   c) Compare the last two elements
   d) Swap the first two elements
   Answer: b) Find the smallest element in the array

7. **Which of the following is the worst case time complexity of selection sort?**
   a) O(n)
   b) O(n log n)
   c) O(n^2)
   d) O(2^n)
   Answer: c) O(n^2)

8. **Which of the following is the average case time complexity of selection sort?**
   a) O(n)
   b) O(n log n)
   c) O(n^2)
   d) O(2^n)
   Answer: c) O(n^2)

9. **Which of the following is the last step in selection sort?**
   a) Swap the last two elements
   b) Swap the first two elements
   c) Find the smallest element in the array
   d) Compare the last two elements
   Answer: a) Swap the last two elements

10. **Which of the following is a disadvantage of selection sort?**
    a) It is a very slow algorithm
    b) It is not stable
    c) It requires additional memory space
    d) It cannot handle large datasets
    Answer: a) It is a very slow algorithm

Lec 45 - Divide and Conquer

1. **Which of the following is not an example of Divide and Conquer algorithm?**
   a) Binary Search
   b) QuickSort
   c) Bubble Sort
   d) MergeSort

Answer: c) Bubble Sort

2. **What is the time complexity of QuickSort algorithm?**
   a) O(n)
   b) O(n^2)
   c) O(n log n)
   d) O(log n)

Answer: c) O(n log n)

3. **In MergeSort algorithm, what is the time complexity of merging two sorted arrays of size n?**
   a) O(n)
   b) O(n^2)
   c) O(log n)
   d) O(1)

Answer: a) O(n)

4. **Which of the following is not a step in the Divide and Conquer algorithm?**
   a) Divide
   b) Conquer
   c) Combine
   d) Increment

Answer: d) Increment

5. **Which of the following is an example of a problem that can be solved using Divide and Conquer algorithm?**
   a) Finding the maximum value in an unsorted array
   b) Counting the number of occurrences of a given element in an unsorted array
   c) Sorting an array in ascending order
   d) Finding the shortest path between two nodes in a graph

Answer: c) Sorting an array in ascending order

6. **What is the space complexity of MergeSort algorithm?**
   a) O(n)
   b) O(n^2)
   c) O(log n)
   d) O(1)

Answer: a) O(n)

7. **Which of the following algorithms uses Divide and Conquer approach to find the closest pair of points in a plane?**
   a) Insertion Sort
   b) Selection Sort
   c) MergeSort
   d) Divide and Conquer algorithm for Closest Pair problem

Answer: d) Divide and Conquer algorithm for Closest Pair problem

8. **What is the worst case time complexity of Binary Search algorithm?**
   a) O(1)
   b) O(log n)
   c) O(n)
   d) O(n^2)

Answer: b) O(log n)

9. **Which of the following is an advantage of using Divide and Conquer approach?**
   a) It is easy to implement
   b) It always gives the optimal solution
   c) It reduces the time complexity of the algorithm
   d) It is not affected by the size of the input

Answer: c) It reduces the time complexity of the algorithm

10. **Which of the following is a disadvantage of using Divide and Conquer approach?**
    a) It is not suitable for solving large problems
    b) It requires extra space for storing the intermediate results
    c) It is difficult to understand and implement
    d) It always gives the correct solution

Answer: b) It requires extra space for storing the intermediate results