

CS301

Data Structures

Important mcqs

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 - Huffman 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

- In threaded binary tree, a node that has no left child and whose left pointer points to the _____.**
 - in-order predecessor
 - in-order successor
 - null
 - none of the mentioned**Answer: a**
- What is a threaded binary tree?**
 - A binary tree in which each node can have any number of children
 - A binary tree in which all the left pointers point to inorder predecessors and right pointers point to inorder successors.
 - A binary tree in which each node can have at most 2 children
 - A binary tree in which all the leaf nodes have a level of 0.**Answer: b**
- What is the time complexity for finding the inorder successor in a threaded binary tree?**
 - $O(1)$
 - $O(n)$
 - $O(\log n)$
 - $O(n^2)$**Answer: a**
- In which traversal, the nodes are visited in increasing order of their values?**
 - Inorder Traversal
 - Preorder Traversal
 - Postorder Traversal
 - Level order Traversal**Answer: a**
- In threaded binary trees, the right pointer of a node points to its _____.**
 - Predecessor
 - Successor
 - Ancestor
 - Descendant**Answer: b**
- Which of the following is not true for threaded binary trees?**
 - In-order traversal can be performed in $O(n)$ time complexity
 - They save storage space
 - They are more efficient than normal binary trees for finding in-order predecessors and successors.
 - They allow for easy deletion of a node**Answer: d**
- Which of the following is a disadvantage of threaded binary trees?**
 - They take up more space than normal binary trees
 - They are less efficient than normal binary trees for finding in-order predecessors and successors.
 - 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 - percolateDown 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?
- The problem of finding the minimum number of equivalence classes for a given set of elements.
 - The problem of efficiently maintaining equivalence relations under dynamic changes to a set of elements.
 - The problem of finding the maximum number of equivalence classes for a given set of elements.
 - 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?
- Arrays
 - Linked lists
 - Binary search trees
 - 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?
- $O(1)$
 - $O(\log n)$
 - $O(n)$
 - $O(n \log n)$

Answer: b

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

Answer: c

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

- Which of the following data structures is a probabilistic data structure?**
 - Binary search tree
 - AVL tree
 - Skip list
 - Red-black tree**Answer: C) Skip list**
- In a skip list, what is the maximum number of levels that a node can have?**
 - 1
 - 2
 - $\log n$
 - Unlimited**Answer: D) Unlimited**
- What is the time complexity of searching for an element in a skip list?**
 - $O(n)$
 - $O(\log n)$
 - $O(n \log n)$
 - $O(1)$**Answer: B) $O(\log n)$**
- Which of the following operations cannot be performed on a skip list?**
 - Insertion
 - Deletion
 - Searching
 - Sorting**Answer: D) Sorting**
- Which of the following is the main advantage of using skip lists over balanced trees?**
 - Space efficiency
 - Time efficiency
 - Ease of implementation
 - None of the above**Answer: C) Ease of implementation**
- Which of the following is a disadvantage of using skip lists?**
 - High space complexity
 - High time complexity
 - Limited applicability
 - None of the above**Answer: A) High space complexity**
- In a skip list, what is the probability of a node having $k+1$ levels, given that it has k levels?**
 - $1/2$
 - $1/4$
 - $1/8$
 - $1/16$**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

