### **MCQs on HashMap in Java**

1. **What is the initial capacity of a HashMap in Java by default?**
   * A) 8
   * B) 16
   * C) 32
   * D) 64  
     **Answer:** B) 16
2. **What is the load factor of a HashMap in Java by default?**
   * A) 0.5
   * B) 0.75
   * C) 1.0
   * D) 2.0  
     **Answer:** B) 0.75
3. **Which of the following methods is used to retrieve a value from a HashMap?**
   * A) getValue()
   * B) fetch()
   * C) get()
   * D) retrieve()  
     **Answer:** C) get()
4. **If two keys are equal in a HashMap, what will happen?**
   * A) Both will be stored
   * B) The second will overwrite the first
   * C) An exception will be thrown
   * D) Both will be ignored  
     **Answer:** B) The second will overwrite the first
5. **Which of the following is true about HashMap?**
   * A) It is synchronized.
   * B) It allows null keys and null values.
   * C) It maintains the order of elements.
   * D) It does not allow duplicate keys.  
     **Answer:** B) It allows null keys and null values.
6. **How can you iterate over the keys in a HashMap?**
   * A) Using keySet()
   * B) Using values()
   * C) Using entrySet()
   * D) Using iterator()  
     **Answer:** A) Using keySet()
7. **Which of the following is the time complexity for the get() operation in a HashMap?**
   * A) O(1) on average
   * B) O(n)
   * C) O(log n)
   * D) O(n^2)  
     **Answer:** A) O(1) on average
8. **What will happen if you try to add a duplicate key to a HashMap?**
   * A) It will throw an exception.
   * B) It will overwrite the existing key's value.
   * C) It will ignore the new entry.
   * D) It will create a new entry with a different key.  
     **Answer:** B) It will overwrite the existing key's value.
9. **Which of the following interfaces does HashMap implement?**
   * A) List
   * B) Set
   * C) Map
   * D) Collection  
     **Answer:** C) Map
10. **What is the maximum number of null keys allowed in a HashMap?**
    * A) 0
    * B) 1
    * C) 2
    * D) Unlimited  
      **Answer:** B) 1
11. **How does a HashMap handle collisions?**
    * A) It throws an exception.
    * B) It uses linked lists (or trees) to store multiple values at the same index.
    * C) It ignores the new entry.
    * D) It merges the keys.  
      **Answer:** B) It uses linked lists (or trees) to store multiple values at the same index.
12. **Which method is used to remove a key-value pair from a HashMap?**
    * A) delete()
    * B) remove()
    * C) discard()
    * D) erase()  
      **Answer:** B) remove()
13. **What happens if you pass a null key to a HashMap?**
    * A) An exception is thrown.
    * B) It will store the null key with a null value.
    * C) It will ignore the entry.
    * D) It will create a new entry with a different key.  
      **Answer:** B) It will store the null key with a null value.
14. **Which of the following methods can be used to get all values from a HashMap?**
    * A) values()
    * B) getValues()
    * C) collectValues()
    * D) retrieveValues()  
      **Answer:** A) values()
15. **What is the effect of the load factor on a HashMap?**
    * A) It determines how many elements can be added.
    * B) It determines when to resize the HashMap.
    * C) It affects the retrieval time.
    * D) It has no effect.  
      **Answer:** B) It determines when to resize the HashMap.
16. **How do you check if a key exists in a HashMap?**
    * A) containsKey()
    * B) hasKey()
    * C) isKeyPresent()
    * D) findKey()  
      **Answer:** A) containsKey()
17. **What is the primary use of the entrySet() method in a HashMap?**
    * A) To get keys only.
    * B) To get values only.
    * C) To get a set of key-value pairs.
    * D) To get the size of the map.  
      **Answer:** C) To get a set of key-value pairs.
18. **Which of the following statements is true about the order of elements in a HashMap?**
    * A) It maintains insertion order.
    * B) It maintains natural order of keys.
    * C) It does not guarantee any order.
    * D) It maintains reverse order of keys.  
      **Answer:** C) It does not guarantee any order.
19. **What is the method used to clear all entries in a HashMap?**
    * A) removeAll()
    * B) clear()
    * C) deleteAll()
    * D) reset()  
      **Answer:** B) clear()
20. **If a HashMap is resized, what happens to the existing entries?**
    * A) They are lost.
    * B) They are rehashed and redistributed.
    * C) They remain unchanged.
    * D) They are converted to a list.  
      **Answer:** B) They are rehashed and redistributed.
21. **Which of the following classes can be used to create a thread-safe version of HashMap?**
    * A) ConcurrentHashMap
    * B) Hashtable
    * C) SynchronizedHashMap
    * D) Both A and B  
      **Answer:** D) Both A and B
22. **What is the primary advantage of using a HashMap?**
    * A) Memory efficiency
    * B) Fast access time
    * C) Maintains order
    * D) Allows duplicates  
      **Answer:** B) Fast access time
23. **Which of the following is not a method of HashMap?**
    * A) putAll()
    * B) putIfAbsent()
    * C) add()
    * D) computeIfAbsent()  
      **Answer:** C) add()
24. **What will be the output of new HashMap<>().put(null, "value")?**
    * A) Throws NullPointerException
    * B) Stores the value with null key
    * C) Ignores the entry
    * D) Throws ClassCastException  
      **Answer:** B) Stores the value with null key
25. **How can you get the size of a HashMap?**
    * A) size()
    * B) count()
    * C) getSize()
    * D) length()  
      **Answer:** A) size()
26. **Which of the following is the best way to copy a HashMap?**
    * A) Using clone()
    * B) Using copy constructor
    * C) Using copy() method
    * D) Using assignment operator  
      **Answer:** B) Using copy constructor
27. **What will happen if the number of entries in a HashMap exceeds the product of its load factor and current capacity?**
    * A) The HashMap will throw an exception.
    * B) The HashMap will be resized.
    * C) The HashMap will ignore new entries.
    * D) The HashMap will slow down.  
      **Answer:** B) The HashMap will be resized.
28. **Which method can be used to replace a value in a HashMap?**
    * A) update()
    * B) replace()
    * C) modify()
    * D) change()  
      **Answer:** B) replace()
29. **Which of the following collections can HashMap not use as a key?**
    * A) String
    * B) Integer
    * C) ArrayList
    * D) Custom objects (if equals() and hashCode() are overridden)  
      **Answer:** C) ArrayList
30. **What will the following code return: new HashMap<>().put("key", "value");?**
    * A) null
    * B) "value"
    * C) "key"
    * D) Throws an error  
      **Answer:** A) null

### **MCQs on Sorting and Searching in Java**

1. **Which of the following is a stable sorting algorithm?**
   * A) Quick Sort
   * B) Heap Sort
   * C) Merge Sort
   * D) Selection Sort  
     **Answer:** C) Merge Sort
2. **What is the time complexity of the binary search algorithm?**
   * A) O(n)
   * B) O(log n)
   * C) O(n log n)
   * D) O(n^2)  
     **Answer:** B) O(log n)
3. **Which sorting algorithm divides the array into smaller sub-arrays and sorts them?**
   * A) Insertion Sort
   * B) Merge Sort
   * C) Selection Sort
   * D) Bubble Sort  
     **Answer:** B) Merge Sort
4. **Which of the following sorting algorithms is not in-place?**
   * A) Quick Sort
   * B) Merge Sort
   * C) Insertion Sort
   * D) Heap Sort  
     **Answer:** B) Merge Sort
5. **What is the worst-case time complexity of Quick Sort?**
   * A) O(n)
   * B) O(log n)
   * C) O(n log n)
   * D) O(n^2)  
     **Answer:** D) O(n^2)
6. **Which method of the Arrays class can be used for binary search in Java?**
   * A) binarySearch()
   * B) search()
   * C) find()
   * D) locate()  
     **Answer:** A) binarySearch()
7. **What is the best case time complexity of Bubble Sort?**
   * A) O(n)
   * B) O(log n)
   * C) O(n log n)
   * D) O(n^2)  
     **Answer:** A) O(n)
8. **Which sorting algorithm is generally the fastest for small datasets?**
   * A) Merge Sort
   * B) Quick Sort
   * C) Insertion Sort
   * D) Heap Sort  
     **Answer:** C) Insertion Sort
9. **What type of data structure does the Heap Sort algorithm use?**
   * A) Tree
   * B) Array
   * C) List
   * D) Graph  
     **Answer:** A) Tree
10. **In which sorting algorithm do we select the smallest element from an unsorted array and move it to the beginning?**
    * A) Merge Sort
    * B) Quick Sort
    * C) Selection Sort
    * D) Insertion Sort  
      **Answer:** C) Selection Sort
11. **Which of the following is an application of binary search?**
    * A) Finding the maximum element
    * B) Finding an element in a sorted array
    * C) Sorting an array
    * D) Reversing an array  
      **Answer:** B) Finding an element in a sorted array
12. **Which sorting algorithm repeatedly steps through the list to be sorted, compares adjacent elements, and swaps them if they are in the wrong order?**
    * A) Quick Sort
    * B) Bubble Sort
    * C) Selection Sort
    * D) Merge Sort  
      **Answer:** B) Bubble Sort
13. **What is the average-case time complexity of Quick Sort?**
    * A) O(n)
    * B) O(log n)
    * C) O(n log n)
    * D) O(n^2)  
      **Answer:** C) O(n log n)
14. **Which sorting algorithm uses the concept of divide and conquer?**
    * A) Insertion Sort
    * B) Selection Sort
    * C) Merge Sort
    * D) Both B and C  
      **Answer:** C) Merge Sort
15. **What is the space complexity of Merge Sort?**
    * A) O(1)
    * B) O(n)
    * C) O(log n)
    * D) O(n log n)  
      **Answer:** B) O(n)
16. **Which of the following algorithms is used to find the kth smallest/largest element in an unsorted array?**
    * A) Binary Search
    * B) Quick Select
    * C) Linear Search
    * D) Selection Sort  
      **Answer:** B) Quick Select
17. **In which sorting algorithm does the pivot play a crucial role?**
    * A) Selection Sort
    * B) Bubble Sort
    * C) Quick Sort
    * D) Merge Sort  
      **Answer:** C) Quick Sort
18. **What is the best case time complexity for Insertion Sort?**
    * A) O(n)
    * B) O(n log n)
    * C) O(n^2)
    * D) O(log n)  
      **Answer:** A) O(n)
19. **Which of the following is true for a stable sorting algorithm?**
    * A) It maintains the relative order of records with equal keys.
    * B) It sorts elements faster than unstable sorts.
    * C) It is always implemented using recursion.
    * D) It cannot sort arrays with duplicate values.  
      **Answer:** A) It maintains the relative order of records with equal keys.
20. **Which sorting algorithm is based on a binary tree data structure?**
    * A) Bubble Sort
    * B) Heap Sort
    * C) Merge Sort
    * D) Insertion Sort  
      **Answer:** B) Heap Sort
21. **What is the time complexity of the linear search algorithm?**
    * A) O(1)
    * B) O(log n)
    * C) O(n)
    * D) O(n log n)  
      **Answer:** C) O(n)
22. **Which of the following sorting algorithms has the worst performance on a nearly sorted list?**
    * A) Insertion Sort
    * B) Merge Sort
    * C) Quick Sort
    * D) Selection Sort  
      **Answer:** D) Selection Sort
23. **In the context of sorting, what does "in-place" mean?**
    * A) The algorithm uses additional memory for sorting.
    * B) The algorithm sorts the elements without using extra space.
    * C) The algorithm requires a second array for sorting.
    * D) The algorithm cannot sort large datasets.  
      **Answer:** B) The algorithm sorts the elements without using extra space.
24. **Which of the following sorting algorithms can be implemented with a recursive approach?**
    * A) Bubble Sort
    * B) Selection Sort
    * C) Quick Sort
    * D) Both B and C  
      **Answer:** C) Quick Sort
25. **Which method is used to sort an array in ascending order in Java?**
    * A) Arrays.sort()
    * B) Arrays.order()
    * C) Arrays.arrange()
    * D) Arrays.sequence()  
      **Answer:** A) Arrays.sort()
26. **What is the primary disadvantage of the Selection Sort algorithm?**
    * A) It is not stable.
    * B) It is slow on large lists.
    * C) It requires additional memory.
    * D) It is complex to implement.  
      **Answer:** B) It is slow on large lists.
27. **Which sorting algorithm works by repeatedly merging smaller sorted arrays?**
    * A) Bubble Sort
    * B) Merge Sort
    * C) Insertion Sort
    * D) Quick Sort  
      **Answer:** B) Merge Sort
28. **What type of search algorithm would you use for an unsorted array?**
    * A) Binary Search
    * B) Linear Search
    * C) Interpolation Search
    * D) Ternary Search  
      **Answer:** B) Linear Search
29. **Which of the following is a characteristic of a binary search tree (BST)?**
    * A) The left subtree contains nodes with keys less than the node's key.
    * B) The right subtree contains nodes with keys greater than the node's key.
    * C) Both A and B
    * D) None of the above  
      **Answer:** C) Both A and B
30. **What is the main advantage of using Quick Sort over other sorting algorithms?**
    * A) It is stable.
    * B) It requires less memory.
    * C) It always performs better than O(n log n).
    * D) It has the lowest worst-case time complexity.  
      **Answer:** B) It requires less memory.
31. **What is the best-case time complexity of Merge Sort?**
    * A) O(n)
    * B) O(log n)
    * C) O(n log n)
    * D) O(n^2)  
      **Answer:** C) O(n log n)
32. **Which searching algorithm is more efficient for large, sorted datasets?**
    * A) Linear Search
    * B) Binary Search
    * C) Jump Search
    * D) Interpolation Search  
      **Answer:** B) Binary Search
33. **Which of the following is true about Heap Sort?**
    * A) It is not in-place.
    * B) It is a stable sorting algorithm.
    * C) It has a worst-case time complexity of O(n log n).
    * D) It is faster than Quick Sort in all cases.  
      **Answer:** C) It has a worst-case time complexity of O(n log n).
34. **Which of the following sorting algorithms is adaptive?**
    * A) Quick Sort
    * B) Bubble Sort
    * C) Heap Sort
    * D) Selection Sort  
      **Answer:** B) Bubble Sort
35. **What is the space complexity of Insertion Sort?**
    * A) O(1)
    * B) O(n)
    * C) O(log n)
    * D) O(n log n)  
      **Answer:** A) O(1)
36. **Which sorting algorithm involves partitioning the array around a pivot?**
    * A) Insertion Sort
    * B) Quick Sort
    * C) Bubble Sort
    * D) Selection Sort  
      **Answer:** B) Quick Sort
37. **How does the Interpolation Search algorithm work?**
    * A) It divides the array into two halves.
    * B) It estimates the position of the desired value based on its value.
    * C) It searches sequentially.
    * D) It uses a hash table for searching.  
      **Answer:** B) It estimates the position of the desired value based on its value.
38. **Which of the following algorithms is the best choice for sorting linked lists?**
    * A) Quick Sort
    * B) Merge Sort
    * C) Bubble Sort
    * D) Insertion Sort  
      **Answer:** B) Merge Sort
39. **Which sorting algorithm is often implemented using a tree structure?**
    * A) Quick Sort
    * B) Bubble Sort
    * C) Heap Sort
    * D) Selection Sort  
      **Answer:** C) Heap Sort
40. **What is the time complexity of finding an element in a balanced binary search tree?**
    * A) O(1)
    * B) O(n)
    * C) O(log n)
    * D) O(n log n)  
      **Answer:** C) O(log n)
41. **Which algorithm is based on the divide and conquer strategy?**
    * A) Heap Sort
    * B) Bubble Sort
    * C) Merge Sort
    * D) Selection Sort  
      **Answer:** C) Merge Sort
42. **In which scenario is Linear Search preferred over Binary Search?**
    * A) When the array is sorted.
    * B) When the array is unsorted.
    * C) When the array is very large.
    * D) When duplicates are present.  
      **Answer:** B) When the array is unsorted.
43. **What is the main disadvantage of the Binary Search algorithm?**
    * A) It requires a sorted array.
    * B) It is slow.
    * C) It is complex to implement.
    * D) It cannot handle duplicates.  
      **Answer:** A) It requires a sorted array.
44. **Which sorting algorithm can be implemented using a recursive function?**
    * A) Insertion Sort
    * B) Selection Sort
    * C) Heap Sort
    * D) Merge Sort  
      **Answer:** D) Merge Sort
45. **What is the time complexity of counting sort?**
    * A) O(n)
    * B) O(n log n)
    * C) O(log n)
    * D) O(n^2)  
      **Answer:** A) O(n)
46. **Which of the following is not a comparison-based sorting algorithm?**
    * A) Quick Sort
    * B) Merge Sort
    * C) Counting Sort
    * D) Bubble Sort  
      **Answer:** C) Counting Sort
47. **In Quick Sort, what is the role of the pivot element?**
    * A) It determines the size of the subarrays.
    * B) It is the first element of the array.
    * C) It helps to partition the array.
    * D) It is ignored during the sort.  
      **Answer:** C) It helps to partition the array.
48. **Which of the following methods can be used to implement a priority queue?**
    * A) Array
    * B) Linked List
    * C) Heap
    * D) All of the above  
      **Answer:** C) Heap
49. **What is the main advantage of using a binary search tree over a linear array for searching?**
    * A) It is easier to implement.
    * B) It allows for dynamic resizing.
    * C) It provides faster search times.
    * D) It is simpler to understand.  
      **Answer:** C) It provides faster search times.
50. **Which of the following algorithms has the best average-case performance?**
    * A) Bubble Sort
    * B) Quick Sort
    * C) Selection Sort
    * D) Insertion Sort  
      **Answer:** B) Quick Sort
51. **In the context of binary search trees, what is a leaf node?**
    * A) A node with two children.
    * B) A node with one child.
    * C) A node with no children.
    * D) A node with the highest value.  
      **Answer:** C) A node with no children.
52. **What is the main characteristic of a balanced binary search tree?**
    * A) It is always full.
    * B) The height is kept to a minimum.
    * C) It allows duplicate keys.
    * D) It requires additional space.  
      **Answer:** B) The height is kept to a minimum.
53. **Which of the following algorithms is guaranteed to run in O(n log n) time?**
    * A) Bubble Sort
    * B) Quick Sort (worst-case)
    * C) Merge Sort
    * D) Selection Sort  
      **Answer:** C) Merge Sort
54. **What is the key idea behind the binary search algorithm?**
    * A) Divide the array into two equal halves.
    * B) Eliminate half of the remaining elements at each step.
    * C) Sort the array before searching.
    * D) Search for every element sequentially.  
      **Answer:** B) Eliminate half of the remaining elements at each step.
55. **Which of the following is not a characteristic of the Quick Sort algorithm?**
    * A) It is in-place.
    * B) It is stable.
    * C) It can be implemented recursively.
    * D) It works well on average.  
      **Answer:** B) It is stable.
56. **What type of data does Counting Sort work best with?**
    * A) Large datasets with large ranges of values.
    * B) Small ranges of integer keys.
    * C) Non-numeric data.
    * D) Randomly ordered datasets.  
      **Answer:** B) Small ranges of integer keys.
57. **Which of the following algorithms can be used to sort a collection in Java?**
    * A) Collections.sort()
    * B) Arrays.sort()
    * C) Both A and B
    * D) None of the above  
      **Answer:** C) Both A and B
58. **In which scenario would you prefer Merge Sort over Quick Sort?**
    * A) When memory space is limited.
    * B) When the dataset is very large.
    * C) When stability is required.
    * D) When the array is nearly sorted.  
      **Answer:** C) When stability is required.
59. **Which of the following is true about Hashing?**
    * A) It uses a linear search for retrieving data.
    * B) It uses a key to map to a value.
    * C) It is a sorting algorithm.
    * D) It guarantees a unique key for each value.  
      **Answer:** B) It uses a key to map to a value.
60. **What is the space complexity of binary search?**
    * A) O(1)
    * B) O(n)
    * C) O(log n)
    * D) O(n log n)  
      **Answer:** A) O(1)

### **MCQs on Heaps in Java**

1. **What is the primary use of a heap data structure?**
   * A) Storing data in a sorted order
   * B) Implementing priority queues
   * C) Searching for elements
   * D) Performing quick sort  
     **Answer:** B) Implementing priority queues
2. **Which type of heap is used to implement a max-heap?**
   * A) Complete binary tree
   * B) Binary search tree
   * C) AVL tree
   * D) Red-black tree  
     **Answer:** A) Complete binary tree
3. **In a max-heap, the value of each node is greater than or equal to the values of its children.**
   * A) True
   * B) False  
     **Answer:** A) True
4. **Which class in Java provides a way to implement a priority queue using a heap?**
   * A) ArrayList
   * B) PriorityQueue
   * C) LinkedList
   * D) HashMap  
     **Answer:** B) PriorityQueue
5. **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)  
     **Answer:** B) O(log n)
6. **What is the time complexity for deleting the maximum element from a max-heap?**
   * A) O(1)
   * B) O(log n)
   * C) O(n)
   * D) O(n log n)  
     **Answer:** B) O(log n)
7. **Which of the following operations can be performed in O(log n) time in a heap?**
   * A) Insertion
   * B) Deletion
   * C) Both A and B
   * D) None of the above  
     **Answer:** C) Both A and B
8. **What happens to the structure of a heap when the maximum element is removed?**
   * A) The last element is placed at the root.
   * B) The heap remains unchanged.
   * C) The second maximum becomes the new root.
   * D) The tree is rebalanced.  
     **Answer:** A) The last element is placed at the root.
9. **How many children does a binary heap node have at most?**
   * A) One
   * B) Two
   * C) Three
   * D) Four  
     **Answer:** B) Two
10. **Which of the following best describes a min-heap?**
    * A) Parent nodes are always greater than their children.
    * B) Parent nodes are always less than their children.
    * C) Nodes can have more than two children.
    * D) It is not a complete binary tree.  
      **Answer:** B) Parent nodes are always less than their children.
11. **What is the height of a complete binary heap with n nodes?**
    * A) log(n)
    * B) n
    * C) log(n) + 1
    * D) n + 1  
      **Answer:** C) log(n) + 1
12. **Which method of the PriorityQueue class can be used to retrieve and remove the head of the queue?**
    * A) get()
    * B) peek()
    * C) poll()
    * D) remove()  
      **Answer:** C) poll()
13. **What is the underlying data structure used by Java's PriorityQueue?**
    * A) Array
    * B) Linked List
    * C) Hash Table
    * D) Binary Tree  
      **Answer:** A) Array
14. **In which scenario would you prefer using a min-heap?**
    * A) When you need to find the maximum element quickly.
    * B) When you need to find the minimum element quickly.
    * C) When you need to sort elements in descending order.
    * D) When you need to implement a stack.  
      **Answer:** B) When you need to find the minimum element quickly.
15. **What is the purpose of the heapify process?**
    * A) To sort the heap.
    * B) To maintain the heap property after an insertion or deletion.
    * C) To balance the tree.
    * D) To find the height of the heap.  
      **Answer:** B) To maintain the heap property after an insertion or deletion.
16. **Which of the following is not a characteristic of heaps?**
    * A) Complete binary tree
    * B) Ordered structure
    * C) Used for priority queues
    * D) Must be balanced  
      **Answer:** D) Must be balanced
17. **In a max-heap, which of the following holds true for a parent node?**
    * A) It is less than both its children.
    * B) It is greater than or equal to both its children.
    * C) It is equal to one of its children.
    * D) It can have one or zero children.  
      **Answer:** B) It is greater than or equal to both its children.
18. **How do you build a heap from an unordered array in Java?**
    * A) Insert each element one by one.
    * B) Use the heapify method.
    * C) Sort the array first.
    * D) Use a linked list.  
      **Answer:** B) Use the heapify method.
19. **What is the time complexity of building a heap from an unordered array?**
    * A) O(n)
    * B) O(n log n)
    * C) O(log n)
    * D) O(n^2)  
      **Answer:** A) O(n)
20. **Which algorithm can be implemented using heaps for sorting?**
    * A) Merge Sort
    * B) Quick Sort
    * C) Heap Sort
    * D) Insertion Sort  
      **Answer:** C) Heap Sort
21. **In Java, what method would you use to check the size of a PriorityQueue?**
    * A) length()
    * B) size()
    * C) count()
    * D) getSize()  
      **Answer:** B) size()
22. **What is the main disadvantage of using a binary heap compared to other data structures?**
    * A) It uses too much memory.
    * B) It does not allow duplicate values.
    * C) It is not efficient for searching arbitrary elements.
    * D) It cannot be implemented in Java.  
      **Answer:** C) It is not efficient for searching arbitrary elements.
23. **What is the effect of calling the clear() method on a PriorityQueue in Java?**
    * A) It removes the highest priority element.
    * B) It empties the queue.
    * C) It resets the queue's capacity.
    * D) It does nothing.  
      **Answer:** B) It empties the queue.
24. **Which of the following is true about heap sort?**
    * A) It is a stable sort.
    * B) It is not in-place.
    * C) It is based on the properties of binary heaps.
    * D) It requires O(n^2) time complexity.  
      **Answer:** C) It is based on the properties of binary heaps.
25. **In a binary heap, the parent of a node at index i can be found at which index?**
    * A) (i - 1) / 2
    * B) (i + 1) / 2
    * C) i / 2
    * D) 2 \* i  
      **Answer:** A) (i - 1) / 2
26. **In a binary heap, the children of a node at index i can be found at which indices?**
    * A) 2 \* i + 1 and 2 \* i + 2
    * B) 2 \* i and 2 \* i + 1
    * C) i - 1 and i + 1
    * D) (i + 1) / 2 and (i + 2) / 2  
      **Answer:** A) 2 \* i + 1 and 2 \* i + 2
27. **What is the time complexity of extracting the minimum element from a min-heap?**
    * A) O(1)
    * B) O(log n)
    * C) O(n)
    * D) O(n log n)  
      **Answer:** B) O(log n)
28. **Which of the following scenarios is best suited for using a max-heap?**
    * A) Implementing a scheduling algorithm
    * B) Finding the smallest element
    * C) Sorting elements in ascending order
    * D) Implementing a leaderboard system  
      **Answer:** D) Implementing a leaderboard system
29. **What is the main property that distinguishes a binary heap from other tree structures?**
    * A) It is always balanced.
    * B) It is a complete binary tree.
    * C) It allows duplicate keys.
    * D) It does not allow null values.  
      **Answer:** B) It is a complete binary tree.
30. **What type of heap is used in a priority queue?**
    * A) Binary heap
    * B) Ternary heap
    * C) Fibonacci heap
    * D) All of the above  
      **Answer:** D) All of the above
31. **Which method allows you to insert an element into a PriorityQueue?**
    * A) enqueue()
    * B) add()
    * C) insert()
    * D) push()  
      **Answer:** B) add()
32. **In a min-heap, which operation would yield the minimum element?**
    * A) Removing the last element
    * B) Accessing the root node
    * C) Sorting the heap
    * D) Accessing any leaf node  
      **Answer:** B) Accessing the root node
33. **Which of the following properties does a max-heap maintain?**
    * A) The smallest element is at the root.
    * B) Every parent node is less than its children.
    * C) The largest element is at the root.
    * D) It can have a variable number of children.  
      **Answer:** C) The largest element is at the root.
34. **When using a min-heap for a Dijkstra's algorithm implementation, what is the purpose of the heap?**
    * A) To store visited nodes
    * B) To quickly find the next node with the smallest distance
    * C) To store the final path
    * D) To keep track of the total weight  
      **Answer:** B) To quickly find the next node with the smallest distance
35. **What would be the result of calling the peek() method on a PriorityQueue?**
    * A) It removes the head of the queue.
    * B) It returns the head of the queue without removing it.
    * C) It adds a new element to the queue.
    * D) It empties the queue.  
      **Answer:** B) It returns the head of the queue without removing it.
36. **In the context of heaps, what does "heapify" mean?**
    * A) Sorting the heap
    * B) Transforming a complete binary tree into a heap
    * C) Merging two heaps
    * D) Finding the maximum element  
      **Answer:** B) Transforming a complete binary tree into a heap
37. **What is the result of inserting elements into a max-heap?**
    * A) The heap remains unchanged.
    * B) The heap property is maintained.
    * C) The smallest element is always at the root.
    * D) The heap becomes unbalanced.  
      **Answer:** B) The heap property is maintained.
38. **Which of the following is an example of a non-binary heap?**
    * A) Ternary heap
    * B) Fibonacci heap
    * C) Both A and B
    * D) None of the above  
      **Answer:** C) Both A and B
39. **How can you remove the minimum element from a min-heap in Java?**
    * A) Using the remove() method
    * B) Using the poll() method
    * C) Using the delete() method
    * D) Using the pop() method  
      **Answer:** B) Using the poll() method
40. **What is the worst-case time complexity for building a binary heap?**
    * A) O(n)
    * B) O(log n)
    * C) O(n log n)
    * D) O(n^2)  
      **Answer:** A) O(n)
41. **Which of the following methods can be used to create a min-heap from an array in Java?**
    * A) Arrays.sort()
    * B) Collections.min()
    * C) PriorityQueue constructor
    * D) Heap.create()  
      **Answer:** C) PriorityQueue constructor
42. **In a min-heap, after deleting the minimum element, what is the next step?**
    * A) Restructure the heap.
    * B) Leave it as is.
    * C) Sort the heap.
    * D) Add a new minimum element.  
      **Answer:** A) Restructure the heap.
43. **What is the time complexity of merging two heaps?**
    * A) O(1)
    * B) O(n)
    * C) O(log n)
    * D) O(n log n)  
      **Answer:** D) O(n log n)
44. **In which scenario would a Fibonacci heap be preferred over a binary heap?**
    * A) When speed is not a concern.
    * B) When the number of insertions is much larger than deletions.
    * C) When memory usage is critical.
    * D) When using a PriorityQueue.  
      **Answer:** B) When the number of insertions is much larger than deletions.
45. **Which of the following statements is true about heaps?**
    * A) Heaps can be implemented using arrays.
    * B) Heaps must always be balanced.
    * C) All heaps are complete binary trees.
    * D) Both A and C  
      **Answer:** D) Both A and C
46. **What is the minimum number of nodes in a complete binary tree of height h?**
    * A) h + 1
    * B) 2^h
    * C) 2^(h+1) - 1
    * D) 2^h - 1  
      **Answer:** A) h + 1
47. **Which of the following can be a consequence of using a binary heap?**
    * A) Increased search time for arbitrary elements.
    * B) Decreased memory usage.
    * C) Easier implementation of sorted lists.
    * D) No duplicates allowed.  
      **Answer:** A) Increased search time for arbitrary elements.
48. **What is the maximum height of a binary heap with n nodes?**
    * A) n
    * B) log(n)
    * C) log(n) + 1
    * D) n + 1  
      **Answer:** C) log(n) + 1
49. **Which of the following can affect the efficiency of heap operations?**
    * A) The type of heap used (min or max)
    * B) The number of elements in the heap
    * C) The implementation of the heap
    * D) All of the above  
      **Answer:** D) All of the above
50. **How do you access the smallest element in a min-heap?**
    * A) By calling the remove() method
    * B) By accessing the root node
    * C) By iterating through all elements
    * D) By sorting the heap  
      **Answer:** B) By accessing the root node
51. **What is the result of inserting a duplicate element into a heap?**
    * A) It is ignored.
    * B) It replaces the existing element.
    * C) It is added as a separate node.
    * D) Both B and C  
      **Answer:** D) Both B and C
52. **Which heap operation has a time complexity of O(n) in the worst case?**
    * A) Insertion
    * B) Deletion
    * C) Building a heap
    * D) Searching for an element  
      **Answer:** C) Building a heap
53. **In a binary heap, if a node has an index i, what are the indices of its children?**
    * A) 2i and 2i + 1
    * B) 2i + 1 and 2i + 2
    * C) i + 1 and i + 2
    * D) i - 1 and i + 1  
      **Answer:** B) 2i + 1 and 2i + 2
54. **Which algorithm is used for heap sort?**
    * A) Selection Sort
    * B) Insertion Sort
    * C) Merge Sort
    * D) Heapify and extract-max  
      **Answer:** D) Heapify and extract-max
55. **What is the key difference between a binary heap and a binary search tree?**
    * A) A binary heap is a complete binary tree, while a binary search tree is not.
    * B) A binary heap maintains order between parent and children, while a binary search tree maintains order between nodes.
    * C) A binary heap is used for priority queues, while a binary search tree is not.
    * D) All of the above  
      **Answer:** D) All of the above
56. **What is the effect of a bubble-up operation in a heap?**
    * A) It maintains the heap property after an insertion.
    * B) It rearranges the heap to be sorted.
    * C) It deletes the maximum element.
    * D) It creates a new heap.  
      **Answer:** A) It maintains the heap property after an insertion.
57. **What is the average-case time complexity of accessing the maximum element in a max-heap?**
    * A) O(1)
    * B) O(log n)
    * C) O(n)
    * D) O(n log n)  
      **Answer:** A) O(1)
58. **Which heap type allows for more efficient merging operations?**
    * A) Binary heap
    * B) Ternary heap
    * C) Fibonacci heap
    * D) Binomial heap  
      **Answer:** C) Fibonacci heap
59. **What happens if a binary heap becomes unbalanced?**
    * A) It cannot perform any operations.
    * B) It will still function, but operations may be inefficient.
    * C) It becomes a binary search tree.
    * D) It is automatically rebalanced.  
      **Answer:** B) It will still function, but operations may be inefficient.
60. **What is the maximum number of nodes in a binary heap of height h?**
    * A) 2^h
    * B) 2^(h+1) - 1
    * C) 2h
    * D) h^2  
      **Answer:** B) 2^(h+1) - 1

### **MCQs on GCD and Prime Factorization in Java**

1. **What is the GCD of 48 and 18?**
   * A) 6
   * B) 12
   * C) 18
   * D) 24  
     **Answer:** A) 6
2. **Which Java method can be used to calculate the GCD of two integers?**
   * A) Math.gcd()
   * B) Math.hcf()
   * C) Math.gcd(int a, int b)
   * D) Math.lcm(int a, int b)  
     **Answer:** C) Math.gcd(int a, int b)
3. **What is the prime factorization of 60?**
   * A) 2 × 2 × 3 × 5
   * B) 2 × 3 × 10
   * C) 4 × 15
   * D) 3 × 20  
     **Answer:** A) 2 × 2 × 3 × 5
4. **Which of the following algorithms is commonly used to calculate the GCD?**
   * A) Sieve of Eratosthenes
   * B) Euclidean algorithm
   * C) Bubble sort
   * D) Quick sort  
     **Answer:** B) Euclidean algorithm
5. **What is the result of the following Java code snippet? System.out.println(Math.gcd(20, 30));**
   * A) 10
   * B) 20
   * C) 30
   * D) 60  
     **Answer:** A) 10
6. **Which of the following statements is true regarding GCD?**
   * A) GCD can be greater than both numbers.
   * B) GCD is always 1 for prime numbers.
   * C) GCD is always smaller than the smallest number.
   * D) GCD is undefined for negative integers.  
     **Answer:** C) GCD is always smaller than the smallest number.
7. **What is the prime factorization of 100?**
   * A) 2 × 5 × 5
   * B) 4 × 25
   * C) 10 × 10
   * D) 2 × 2 × 5 × 5  
     **Answer:** D) 2 × 2 × 5 × 5
8. **What is the least common multiple (LCM) of two numbers if their GCD is known?**
   * A) (a × b) / GCD(a, b)
   * B) GCD(a, b) / (a × b)
   * C) a + b
   * D) a × b  
     **Answer:** A) (a × b) / GCD(a, b)
9. **How would you implement a method to find the GCD of two numbers using recursion in Java?**
   * A) Use a loop until one number is zero.
   * B) Use the formula GCD(a, b) = GCD(b, a % b).
   * C) Directly return the smaller of the two numbers.
   * D) Calculate the product and divide by the sum.  
     **Answer:** B) Use the formula GCD(a, b) = GCD(b, a % b).
10. **Which of the following correctly finds the prime factors of a number in Java?**
    * A) Divide by all integers until 1.
    * B) Check divisibility by prime numbers only.
    * C) Use a hash map to store factors.
    * D) All of the above.  
      **Answer:** B) Check divisibility by prime numbers only.
11. **If two numbers are coprime, what is their GCD?**
    * A) 0
    * B) 1
    * C) -1
    * D) Both numbers  
      **Answer:** B) 1
12. **What is the prime factorization of 72?**
    * A) 2 × 3 × 12
    * B) 2 × 2 × 2 × 3 × 3
    * C) 2 × 6 × 6
    * D) 4 × 18  
      **Answer:** B) 2 × 2 × 2 × 3 × 3
13. **Which Java package contains the BigInteger class, which can be used for calculating GCD of large integers?**
    * A) java.util
    * B) java.math
    * C) java.lang
    * D) java.io  
      **Answer:** B) java.math
14. **In the context of prime factorization, what is the significance of a prime number?**
    * A) It can only be divided by 1 and itself.
    * B) It can be expressed as a product of other numbers.
    * C) It is always even.
    * D) It has more than two factors.  
      **Answer:** A) It can only be divided by 1 and itself.
15. **What is the GCD of 15, 45, and 75?**
    * A) 5
    * B) 15
    * C) 30
    * D) 45  
      **Answer:** B) 15

### **MCQs on Binary Trees and Binary Search Trees in Java**

1. **What is the maximum number of nodes in a binary tree of height h?**
   * A) 2h−12^h - 12h−1
   * B) 2h+1−12^{h+1} - 12h+1−1
   * C) h2h^2h2
   * D) h+1h + 1h+1  
     **Answer:** B) 2h+1−12^{h+1} - 12h+1−1
2. **Which traversal method visits the root node first?**
   * A) In-order
   * B) Pre-order
   * C) Post-order
   * D) Level-order  
     **Answer:** B) Pre-order
3. **What is the key property of a binary search tree (BST)?**
   * A) All nodes have two children.
   * B) Left child < Parent < Right child.
   * C) All values are unique.
   * D) Both B and C.  
     **Answer:** D) Both B and C.
4. **Which of the following is not a valid binary tree traversal method?**
   * A) In-order
   * B) Pre-order
   * C) Post-order
   * D) Cross-order  
     **Answer:** D) Cross-order
5. **What is the height of a binary tree with a single node?**
   * A) 0
   * B) 1
   * C) 2
   * D) Undefined  
     **Answer:** A) 0
6. **Which Java class is commonly used to implement a binary tree?**
   * A) ArrayList
   * B) HashMap
   * C) LinkedList
   * D) A custom class with nodes  
     **Answer:** D) A custom class with nodes
7. **In a BST, the left subtree of a node contains values that are:**
   * A) Greater than the node's value.
   * B) Less than or equal to the node's value.
   * C) Less than the node's value.
   * D) Randomly ordered.  
     **Answer:** C) Less than the node's value.
8. **How do you find the minimum value in a BST?**
   * A) Go to the rightmost node.
   * B) Go to the leftmost node.
   * C) The root node is always the minimum.
   * D) Use a queue.  
     **Answer:** B) Go to the leftmost node.
9. **What is the time complexity of searching for an element in a balanced BST?**
   * A) O(1)
   * B) O(log n)
   * C) O(n)
   * D) O(n log n)  
     **Answer:** B) O(log n)
10. **Which traversal method produces the values of a BST in sorted order?**
    * A) Pre-order
    * B) In-order
    * C) Post-order
    * D) Level-order  
      **Answer:** B) In-order
11. **What is the maximum number of children a binary tree node can have?**
    * A) One
    * B) Two
    * C) Three
    * D) Unlimited  
      **Answer:** B) Two
12. **Which of the following is true about a complete binary tree?**
    * A) All levels are fully filled except possibly the last.
    * B) All nodes have two children.
    * C) The last level is filled from right to left.
    * D) None of the above.  
      **Answer:** A) All levels are fully filled except possibly the last.
13. **What does the depth of a binary tree represent?**
    * A) The number of nodes in the tree.
    * B) The height of the tree.
    * C) The number of edges from the root to a node.
    * D) The total number of leaf nodes.  
      **Answer:** C) The number of edges from the root to a node.
14. **What is the average-case time complexity for inserting an element into a BST?**
    * A) O(1)
    * B) O(log n)
    * C) O(n)
    * D) O(n log n)  
      **Answer:** B) O(log n)
15. **Which of the following structures can be used to represent a binary tree node in Java?**
    * A) Array
    * B) LinkedList
    * C) Custom class with left and right references
    * D) HashMap  
      **Answer:** C) Custom class with left and right references
16. **In a binary tree, what is a leaf node?**
    * A) A node with one child.
    * B) A node with two children.
    * C) A node with no children.
    * D) The root node.  
      **Answer:** C) A node with no children.
17. **What is the height of a complete binary tree with n nodes?**
    * A) log(n)
    * B) n
    * C) log(n) + 1
    * D) n + 1  
      **Answer:** C) log(n) + 1
18. **Which of the following is not a property of binary search trees?**
    * A) Left subtree contains values less than the parent.
    * B) Right subtree contains values greater than the parent.
    * C) All nodes must have two children.
    * D) No duplicate values are allowed.  
      **Answer:** C) All nodes must have two children.
19. **Which Java method is commonly used to perform in-order traversal of a binary tree?**
    * A) traverseInOrder()
    * B) inOrder()
    * C) printInOrder()
    * D) None of the above (must be implemented)  
      **Answer:** D) None of the above (must be implemented)
20. **In a binary tree, the term "sibling" refers to:**
    * A) Parent nodes.
    * B) Nodes that share the same parent.
    * C) Leaf nodes.
    * D) All nodes.  
      **Answer:** B) Nodes that share the same parent.
21. **What is the result of a post-order traversal on a binary tree?**
    * A) Left, Right, Root
    * B) Root, Left, Right
    * C) Right, Left, Root
    * D) Left, Root, Right  
      **Answer:** A) Left, Right, Root
22. **Which traversal method is most suitable for copying a binary tree?**
    * A) Pre-order
    * B) In-order
    * C) Post-order
    * D) Level-order  
      **Answer:** A) Pre-order
23. **What is the time complexity for deleting a node in a BST in the average case?**
    * A) O(1)
    * B) O(log n)
    * C) O(n)
    * D) O(n log n)  
      **Answer:** B) O(log n)
24. **How do you determine if a binary tree is a valid BST?**
    * A) Check if all values are unique.
    * B) Ensure in-order traversal yields sorted values.
    * C) Count the number of nodes.
    * D) All of the above.  
      **Answer:** B) Ensure in-order traversal yields sorted values.
25. **Which property holds true for the ancestor nodes in a binary search tree?**
    * A) They are always less than their descendants.
    * B) They can be equal to their descendants.
    * C) They are always greater than their descendants.
    * D) Both A and C.  
      **Answer:** D) Both A and C.
26. **What will happen if you attempt to insert a duplicate value into a BST?**
    * A) The value is ignored.
    * B) An exception is thrown.
    * C) The tree becomes invalid.
    * D) It creates a new node.  
      **Answer:** A) The value is ignored.
27. **Which of the following is not a balanced binary tree?**
    * A) AVL tree
    * B) Red-black tree
    * C) Complete binary tree
    * D) Simple binary tree  
      **Answer:** D) Simple binary tree
28. **What is the main advantage of using a BST over a regular binary tree?**
    * A) Easier to implement.
    * B) Faster search, insertion, and deletion operations.
    * C) Uses less memory.
    * D) Always balanced.  
      **Answer:** B) Faster search, insertion, and deletion operations.
29. **What will be the result of the in-order traversal of a BST containing the values [10, 5, 15]?**
    * A) [10, 5, 15]
    * B) [5, 10, 15]
    * C) [15, 10, 5]
    * D) [10, 15, 5]  
      **Answer:** B) [5, 10, 15]
30. **How can you identify a leaf node in a binary tree using Java?**
    * A) A node with one child.
    * B) A node with two children.
    * C) A node that has no left or right child.
    * D) A node that is the last in level-order.  
      **Answer:** C) A node that has no left or right child.
31. **In a binary tree, which node is referred to as the "root"?**
    * A) The first node in any traversal.
    * B) The topmost node.
    * C) Any node with no children.
    * D) The node with the highest value.  
      **Answer:** B) The topmost node.
32. **Which data structure is typically used to implement the level-order traversal of a binary tree?**
    * A) Stack
    * B) Queue
    * C) LinkedList
    * D) Array  
      **Answer:** B) Queue
33. **What is the purpose of a self-balancing BST?**
    * A) To keep nodes in sorted order.
    * B) To ensure that the height of the tree remains logarithmic.
    * C) To make all nodes have two children.
    * D) To allow for faster access.  
      **Answer:** B) To ensure that the height of the tree remains logarithmic.
34. **In a binary tree, what is the relationship between the number of leaf nodes and the number of internal nodes?**
    * A) Leaf nodes are always greater than internal nodes.
    * B) The number of internal nodes is equal to the number of leaf nodes.
    * C) For a complete binary tree, the number of leaf nodes is always one more than the number of internal nodes.
    * D) There is no specific relationship.  
      **Answer:** C) For a complete binary tree, the number of leaf nodes is always one more than the number of internal nodes.
35. **What will happen if you perform an in-order traversal on a non-BST?**
    * A) It will produce sorted output.
    * B) It will produce unsorted output.
    * C) It will throw an error.
    * D) It will be the same as level-order traversal.  
      **Answer:** B) It will produce unsorted output.
36. **In a BST, if a node has two children, which node will replace it when it is deleted?**
    * A) The left child.
    * B) The right child.
    * C) The maximum node from the left subtree or the minimum node from the right subtree.
    * D) None of the above.  
      **Answer:** C) The maximum node from the left subtree or the minimum node from the right subtree.
37. **What is the minimum number of nodes in a binary tree of height h?**
    * A) h+1h + 1h+1
    * B) 2h2^h2h
    * C) 2h+1−12^{h+1} - 12h+1−1
    * D) h2h^2h2  
      **Answer:** A) h+1h + 1h+1
38. **Which property is specific to AVL trees?**
    * A) They are always complete.
    * B) They maintain a height balance property.
    * C) They can have duplicate values.
    * D) They are implemented using linked lists.  
      **Answer:** B) They maintain a height balance property.
39. **How can you implement a binary tree in Java?**
    * A) By using built-in Java collections.
    * B) By creating a Node class and linking them.
    * C) By using an array.
    * D) Both B and C.  
      **Answer:** D) Both B and C.
40. **What will be the in-order traversal result of a BST containing nodes with values 3, 1, 4, 2?**
    * A) [3, 1, 4, 2]
    * B) [1, 2, 3, 4]
    * C) [4, 3, 2, 1]
    * D) [2, 1, 3, 4]  
      **Answer:** B) [1, 2, 3, 4]
41. **What is the main advantage of using a BST over an unsorted array for search operations?**
    * A) Simplicity of implementation.
    * B) Faster average-case search time.
    * C) Less memory usage.
    * D) All of the above.  
      **Answer:** B) Faster average-case search time.
42. **How can you check if a binary tree is balanced?**
    * A) Check if all nodes have two children.
    * B) Ensure the difference in height between left and right subtrees is at most 1.
    * C) Count the total number of nodes.
    * D) Perform level-order traversal.  
      **Answer:** B) Ensure the difference in height between left and right subtrees is at most 1.
43. **Which of the following statements is true regarding the height of a binary search tree?**
    * A) It can be equal to the number of nodes.
    * B) It is always less than the number of nodes.
    * C) It is the same as the depth of the tree.
    * D) It cannot be determined without traversing the tree.  
      **Answer:** A) It can be equal to the number of nodes.
44. **What is the depth of the root node in a binary tree?**
    * A) 0
    * B) 1
    * C) Equal to the height of the tree.
    * D) Undefined.  
      **Answer:** A) 0
45. **What traversal would you use to clone a binary tree?**
    * A) Pre-order traversal.
    * B) In-order traversal.
    * C) Post-order traversal.
    * D) Level-order traversal.  
      **Answer:** A) Pre-order traversal.
46. **Which of the following will result in a valid binary search tree?**
    * A) Each node has only one child.
    * B) Nodes have random values.
    * C) Each left child is less than the parent and each right child is greater.
    * D) All values are equal.  
      **Answer:** C) Each left child is less than the parent and each right child is greater.
47. **How can you determine the depth of a node in a binary tree?**
    * A) Count the number of edges from the node to the root.
    * B) Count the number of leaf nodes.
    * C) Use the height of the tree.
    * D) Both A and C.  
      **Answer:** A) Count the number of edges from the node to the root.
48. **Which of the following operations is not typically performed on a binary tree?**
    * A) Insert
    * B) Delete
    * C) Rotate
    * D) Union  
      **Answer:** D) Union
49. **In a balanced binary search tree, what is the relationship between the height and the number of nodes?**
    * A) Height is directly proportional to the number of nodes.
    * B) Height is logarithmic with respect to the number of nodes.
    * C) Height is constant.
    * D) There is no relationship.  
      **Answer:** B) Height is logarithmic with respect to the number of nodes.
50. **Which method is commonly used to perform a level-order traversal in Java?**
    * A) Stack
    * B) Queue
    * C) ArrayList
    * D) LinkedList  
      **Answer:** B) Queue
51. **What is the primary disadvantage of a binary search tree?**
    * A) It requires more memory than a linked list.
    * B) It can become unbalanced, leading to O(n) time complexity.
    * C) It is difficult to implement.
    * D) It cannot store duplicate values.  
      **Answer:** B) It can become unbalanced, leading to O(n) time complexity.
52. **In a binary tree, how do you represent a null child?**
    * A) Using a sentinel value.
    * B) By using null in Java.
    * C) By creating an empty node.
    * D) Both B and C.  
      **Answer:** D) Both B and C.

**What will be the output of the following Java code snippet?**java  
Copy code  
TreeNode root = new TreeNode(10);

root.left = new TreeNode(5);

root.right = new TreeNode(15);

System.out.println(root.left.value);

* + A) 10
  + B) 5
  + C) 15
  + D) null  
    **Answer:** B) 5

1. **Which data structure is not typically associated with tree traversal?**
   * A) Queue
   * B) Stack
   * C) LinkedList
   * D) Array  
     **Answer:** C) LinkedList
2. **Which of the following algorithms can be used to check if a binary tree is symmetric?**
   * A) Depth-first search
   * B) Breadth-first search
   * C) Both A and B
   * D) None of the above  
     **Answer:** C) Both A and B
3. **What is the maximum height of a binary tree with n nodes?**
   * A) n
   * B) n/2
   * C) log(n)
   * D) log(n) + 1  
     **Answer:** A) n
4. **In a binary search tree, what does the term "successor" refer to?**
   * A) The largest node in the left subtree.
   * B) The smallest node in the right subtree.
   * C) The parent node.
   * D) The last node visited in traversal.  
     **Answer:** B) The smallest node in the right subtree.
5. **What is the minimum number of nodes in a full binary tree of height h?**
   * A) 2^h - 1
   * B) 2^(h+1) - 1
   * C) 2^h
   * D) h + 1  
     **Answer:** A) 2^h - 1
6. **Which of the following is a property of a red-black tree?**
   * A) Every node is either red or black.
   * B) The root is always black.
   * C) No two red nodes can be adjacent.
   * D) All of the above.  
     **Answer:** D) All of the above.
7. **What is the result of a level-order traversal of the tree containing nodes [4, 2, 6, 1, 3, 5, 7]?**
   * A) [4, 2, 6, 1, 3, 5, 7]
   * B) [1, 2, 3, 4, 5, 6, 7]
   * C) [1, 4, 2, 3, 6, 5, 7]
   * D) [4, 1, 2, 3, 6, 5, 7]  
     **Answer:** A) [4, 2, 6, 1, 3, 5, 7]

### **MCQs on Recursion, Backtracking, and Divide and Conquer**

1. **What is recursion in programming?**
   * A) A method that calls itself.
   * B) A method that loops through an array.
   * C) A method that sorts elements.
   * D) A method that terminates a program.  
     **Answer:** A) A method that calls itself.
2. **Which of the following is a base case in recursion?**
   * A) The condition under which recursion stops.
   * B) The recursive case that calls itself.
   * C) The condition for looping.
   * D) None of the above.  
     **Answer:** A) The condition under which recursion stops.
3. **What does backtracking typically solve?**
   * A) Problems with unique solutions.
   * B) Problems with multiple solutions.
   * C) Sorting problems.
   * D) Graph traversal problems.  
     **Answer:** B) Problems with multiple solutions.
4. **What is the time complexity of binary search using the divide and conquer approach?**
   * A) O(n)
   * B) O(log n)
   * C) O(n log n)
   * D) O(1)  
     **Answer:** B) O(log n)
5. **Which of the following problems can be solved using backtracking?**
   * A) N-Queens problem.
   * B) Sorting an array.
   * C) Finding the maximum element.
   * D) Binary search.  
     **Answer:** A) N-Queens problem.
6. **In divide and conquer, what is the main strategy?**
   * A) Solve subproblems independently.
   * B) Solve the problem in a linear fashion.
   * C) Solve all problems at once.
   * D) None of the above.  
     **Answer:** A) Solve subproblems independently.
7. **Which of the following is an example of a divide and conquer algorithm?**
   * A) Depth-first search.
   * B) Merge sort.
   * C) Quick sort.
   * D) Both B and C.  
     **Answer:** D) Both B and C.
8. **What is a characteristic of backtracking algorithms?**
   * A) They always find the optimal solution.
   * B) They make decisions and undo them if necessary.
   * C) They operate in linear time.
   * D) They require extra memory for storing state.  
     **Answer:** B) They make decisions and undo them if necessary.
9. **In recursive functions, what is the purpose of a return statement?**
   * A) To terminate the program.
   * B) To return control to the calling function.
   * C) To create a loop.
   * D) To initiate recursion.  
     **Answer:** B) To return control to the calling function.
10. **Which of the following problems can be efficiently solved using recursion?**
    * A) Finding factorial of a number.
    * B) Iterating through an array.
    * C) Bubble sort.
    * D) Both B and C.  
      **Answer:** A) Finding factorial of a number.
11. **In backtracking, when do you stop exploring a particular path?**
    * A) When a valid solution is found.
    * B) When no further decisions can be made.
    * C) When the maximum depth is reached.
    * D) Both A and B.  
      **Answer:** D) Both A and B.
12. **What is the space complexity of recursion due to function call stack?**
    * A) O(1)
    * B) O(n)
    * C) O(log n)
    * D) O(n^2)  
      **Answer:** B) O(n)
13. **Which of the following is true about the recursive implementation of the Fibonacci series?**
    * A) It is efficient for large inputs.
    * B) It has exponential time complexity.
    * C) It uses constant space.
    * D) It requires no base case.  
      **Answer:** B) It has exponential time complexity.
14. **What is the goal of the Merge Sort algorithm?**
    * A) To sort an array using linear time.
    * B) To divide the array and sort each part separately.
    * C) To repeatedly swap elements.
    * D) To use backtracking to find the correct order.  
      **Answer:** B) To divide the array and sort each part separately.
15. **In the context of backtracking, what does the "n-queens problem" entail?**
    * A) Placing n queens on a chessboard so they attack each other.
    * B) Placing n queens on a chessboard so they do not attack each other.
    * C) Finding the minimum moves for queens to attack.
    * D) All of the above.  
      **Answer:** B) Placing n queens on a chessboard so they do not attack each other.
16. **How does the Quick Sort algorithm choose a pivot?**
    * A) Always the first element.
    * B) Always the last element.
    * C) Randomly or by a specific rule.
    * D) It does not use a pivot.  
      **Answer:** C) Randomly or by a specific rule.
17. **Which of the following statements about recursion is true?**
    * A) It can lead to stack overflow if not designed carefully.
    * B) It is always more efficient than iteration.
    * C) It requires more memory than iteration.
    * D) Both A and C.  
      **Answer:** D) Both A and C.
18. **In backtracking algorithms, what is a "solution space"?**
    * A) The range of inputs.
    * B) All possible configurations of the solution.
    * C) The final output.
    * D) None of the above.  
      **Answer:** B) All possible configurations of the solution.
19. **Which of the following problems is NOT typically solved by the divide and conquer strategy?**
    * A) Sorting
    * B) Searching
    * C) Graph traversal
    * D) Matrix multiplication  
      **Answer:** C) Graph traversal
20. **What does it mean when an algorithm is described as "optimal"?**
    * A) It uses the least amount of memory.
    * B) It finds the best possible solution.
    * C) It runs in constant time.
    * D) It is always the fastest.  
      **Answer:** B) It finds the best possible solution.
21. **Which data structure is commonly used to implement backtracking algorithms?**
    * A) Queue
    * B) Stack
    * C) Array
    * D) LinkedList  
      **Answer:** B) Stack
22. **In the divide and conquer method, how is the problem typically divided?**
    * A) Into two or more smaller subproblems.
    * B) Into larger subproblems.
    * C) Into random-sized subproblems.
    * D) Into one subproblem.  
      **Answer:** A) Into two or more smaller subproblems.
23. **Which of the following is a classic example of a recursive algorithm?**
    * A) Bubble sort
    * B) Tower of Hanoi
    * C) Breadth-first search
    * D) Prim's algorithm  
      **Answer:** B) Tower of Hanoi
24. **What is the worst-case time complexity of the Quick Sort algorithm?**
    * A) O(n log n)
    * B) O(n^2)
    * C) O(n)
    * D) O(log n)  
      **Answer:** B) O(n^2)
25. **What is the main characteristic of the backtracking approach?**
    * A) It always finds the first solution.
    * B) It avoids exploring paths that are not valid.
    * C) It uses dynamic programming.
    * D) It guarantees the shortest path.  
      **Answer:** B) It avoids exploring paths that are not valid.
26. **Which of the following is an example of a problem that can be solved with recursion but not with iteration?**
    * A) Factorial calculation
    * B) Fibonacci sequence
    * C) Array summation
    * D) All of the above  
      **Answer:** D) All of the above
27. **Which algorithm is an example of a backtracking algorithm?**
    * A) Binary search
    * B) Merge sort
    * C) Sudoku solver
    * D) Quick sort  
      **Answer:** C) Sudoku solver
28. **In which scenario is a recursive solution more intuitive than an iterative one?**
    * A) Sorting a list
    * B) Traversing a tree
    * C) Searching an array
    * D) None of the above  
      **Answer:** B) Traversing a tree
29. **What is the main drawback of using recursion?**
    * A) Complexity
    * B) Inefficiency
    * C) Stack overflow
    * D) Both B and C  
      **Answer:** D) Both B and C.
30. **What is the time complexity of the Merge Sort algorithm?**
    * A) O(n)
    * B) O(n log n)
    * C) O(n^2)
    * D) O(log n)  
      **Answer:** B) O(n log n)
31. **In backtracking, what is a "dead end"?**
    * A) A point where no solutions are possible.
    * B) The final solution found.
    * C) The beginning of the search.
    * D) A path that leads to multiple solutions.  
      **Answer:** A) A point where no solutions are possible.
32. **What is the primary goal of the divide and conquer strategy?**
    * A) Minimize memory usage.
    * B) Maximize speed.
    * C) Simplify the problem-solving process.
    * D) Reduce the problem into smaller, manageable parts.  
      **Answer:** D) Reduce the problem into smaller, manageable parts.
33. **Which of the following techniques is not used in backtracking?**
    * A) State space tree
    * B) Pruning
    * C) Greedy method
    * D) Backtracking  
      **Answer:** C) Greedy method
34. **In a recursive function, what happens if no base case is defined?**
    * A) The function terminates correctly.
    * B) The function leads to infinite recursion.
    * C) The function executes in linear time.
    * D) The function performs a loop.  
      **Answer:** B) The function leads to infinite recursion.
35. **What is the essence of the divide and conquer approach?**
    * A) Solving problems in parallel.
    * B) Solving problems recursively by dividing them into smaller parts.
    * C) Finding solutions without recursion.
    * D) Avoiding function calls.  
      **Answer:** B) Solving problems recursively by dividing them into smaller parts.
36. **Which of the following algorithms can be classified as both recursive and divide and conquer?**
    * A) Selection sort
    * B) Insertion sort
    * C) Merge sort
    * D) Linear search  
      **Answer:** C) Merge sort
37. **What is a common application of backtracking?**
    * A) Sorting
    * B) Pathfinding in a maze
    * C) Searching in a tree
    * D) Graph coloring  
      **Answer:** D) Graph coloring
38. **Which of the following is NOT a property of divide and conquer algorithms?**
    * A) They solve problems by breaking them down into smaller subproblems.
    * B) They combine the results of the subproblems.
    * C) They solve subproblems in a sequential manner.
    * D) They may have overlapping subproblems.  
      **Answer:** D) They may have overlapping subproblems.
39. **Which of the following recursive algorithms has a time complexity of O(n^2)?**
    * A) Fibonacci series
    * B) Merge sort
    * C) Quick sort
    * D) Insertion sort  
      **Answer:** A) Fibonacci series (without memoization)
40. **What does the "divide" step in the divide and conquer approach entail?**
    * A) Merging the results.
    * B) Solving the problem directly.
    * C) Dividing the problem into smaller subproblems.
    * D) None of the above.  
      **Answer:** C) Dividing the problem into smaller subproblems.
41. **Which of the following best describes a "recursive function"?**
    * A) A function that uses loops.
    * B) A function that calls itself directly or indirectly.
    * C) A function that only runs once.
    * D) A function that sorts data.  
      **Answer:** B) A function that calls itself directly or indirectly.
42. **What is the characteristic feature of dynamic programming?**
    * A) It uses a greedy approach.
    * B) It solves overlapping subproblems.
    * C) It avoids recursion.
    * D) It is always optimal.  
      **Answer:** B) It solves overlapping subproblems.
43. **In backtracking, how is "pruning" used?**
    * A) To ignore invalid paths.
    * B) To optimize memory usage.
    * C) To speed up the algorithm.
    * D) To create new paths.  
      **Answer:** A) To ignore invalid paths.
44. **Which of the following is NOT an example of a recursive problem?**
    * A) Factorial calculation
    * B) Fibonacci sequence
    * C) Finding the greatest common divisor (GCD)
    * D) Linear search  
      **Answer:** D) Linear search
45. **What is the principle of optimality in dynamic programming?**
    * A) Optimal solutions to subproblems can lead to an optimal solution of the overall problem.
    * B) Solutions must be constructed in a greedy manner.
    * C) Recursive calls are always efficient.
    * D) None of the above.  
      **Answer:** A) Optimal solutions to subproblems can lead to an optimal solution of the overall problem.
46. **Which of the following algorithms divides the array into two halves?**
    * A) Linear search
    * B) Selection sort
    * C) Quick sort
    * D) Both B and C  
      **Answer:** C) Quick sort
47. **What is the best use case for backtracking?**
    * A) Finding the largest number in an array.
    * B) Solving combinatorial problems.
    * C) Searching in a sorted array.
    * D) Sorting elements.  
      **Answer:** B) Solving combinatorial problems.
48. **Which of the following describes a "leaf node" in recursion?**
    * A) A node that has no children.
    * B) A node that is at the root level.
    * C) A node that always contains data.
    * D) None of the above.  
      **Answer:** A) A node that has no children.
49. **In divide and conquer, what is the "conquer" step?**
    * A) Solving the subproblems.
    * B) Dividing the problems.
    * C) Combining the results.
    * D) All of the above.  
      **Answer:** A) Solving the subproblems.
50. **Which of the following recursive functions does NOT have overlapping subproblems?**
    * A) Fibonacci series
    * B) Factorial calculation
    * C) Longest common subsequence
    * D) All of the above have overlapping subproblems.  
      **Answer:** B) Factorial calculation
51. **What does the depth of recursion refer to?**
    * A) The number of recursive calls made.
    * B) The number of variables used in recursion.
    * C) The maximum depth of the recursion tree.
    * D) The total number of function calls.  
      **Answer:** C) The maximum depth of the recursion tree.
52. **What type of problems typically benefit from the divide and conquer approach?**
    * A) Problems with linear time solutions.
    * B) Problems with exponential growth.
    * C) Problems that can be broken into independent subproblems.
    * D) None of the above.  
      **Answer:** C) Problems that can be broken into independent subproblems.
53. **Which of the following algorithms is an example of dynamic programming rather than divide and conquer?**
    * A) Merge sort
    * B) Fibonacci sequence with memoization
    * C) Quick sort
    * D) Binary search  
      **Answer:** B) Fibonacci sequence with memoization
54. **In backtracking, what does "state" refer to?**
    * A) The current position in the problem-solving process.
    * B) The final solution.
    * C) The beginning of the algorithm.
    * D) None of the above.  
      **Answer:** A) The current position in the problem-solving process.
55. **What is the major limitation of the recursive approach?**
    * A) It is difficult to implement.
    * B) It can lead to high memory consumption.
    * C) It is always slower than iterative solutions.
    * D) It cannot solve complex problems.  
      **Answer:** B) It can lead to high memory consumption.
56. **Which of the following is a common use of the recursive approach in algorithms?**
    * A) Finding the shortest path in a graph.
    * B) Generating permutations of a set.
    * C) Linear searching of data.
    * D) Sorting an array.  
      **Answer:** B) Generating permutations of a set.
57. **In the context of recursive functions, what is a "call stack"?**
    * A) A structure to store non-recursive calls.
    * B) A structure that stores active function calls.
    * C) A type of data storage.
    * D) None of the above.  
      **Answer:** B) A structure that stores active function calls.
58. **How can backtracking be implemented in a programming language like Java?**
    * A) Using loops only.
    * B) By using recursion to explore possibilities.
    * C) Using static variables.
    * D) By manipulating arrays directly.  
      **Answer:** B) By using recursion to explore possibilities.
59. **What happens when a recursive function calls itself multiple times?**
    * A) It results in an infinite loop.
    * B) It may cause stack overflow.
    * C) It optimizes memory usage.
    * D) It runs faster.  
      **Answer:** B) It may cause stack overflow.
60. **Which of the following statements is true regarding recursion and iteration?**
    * A) Recursion is always faster than iteration.
    * B) Iteration is always more readable than recursion.
    * C) Both can be used to solve the same problems.
    * D) Recursion is always more memory efficient.  
      **Answer:** C) Both can be used to solve the same problems.