قائمة الاسئلة

تعقيد خوارزميات - المستوى الرابع -قسم علوم حاسوب - - كلية الحاسوب وتكنولوجيا المعلومات - الفترة - درجة الامتحان (86)

د. غالب الجعفري

1) What is the closed-form formula for the sum of the first nnn natural numbers?
   1) - n(n+1)
   2) + n(n+1)/2
   3) - n^2+n
   4) - (n+1)!
2) The asymptotic performance of an algorithm refers to how it behaves as the problem size increases significantly.
   1) + TRUE.
   2) - FALSE.
3) The worst-case analysis of an algorithm provides an average expected running time.
   1) - TRUE.
   2) + FALSE.
4) Strong induction requires proving the base case for multiple initial values.
   1) + TRUE.
   2) - FALSE.
5) What is the primary purpose of asymptotic performance analysis in algorithms?
   1) - To determine the exact execution time of an algorithm
   2) + To understand how an algorithm scales with input size
   3) - To calculate memory usage of a program
   4) - To find syntax errors in an algorithm
6) What is the worst-case time complexity of Insertion Sort?
   1) - O(n)
   2) - O(n log n)
   3) + $O(n^2)$
   4) - $O(2^n)$
7) In Insertion Sort, what is the role of the key variable?
   1) - It stores the largest element in the array
   2) + It temporarily holds the element being compared and inserted
   3) - It keeps track of the sorted portion of the array
   4) - It determines the total number of swaps
8) Which of the following best describes the best-case time complexity of Insertion Sort?
   1) + O(n)
   2) - O(n log n)
   3) - $O(n^2)$
   4) - O(log n)
9) What is the primary reason Insertion Sort is not suitable for large inputs?
   1) - It requires additional memory space
   2) + It has a high worst-case time complexity
   3) - It is difficult to implement
   4) - It does not work for unordered data
10) Which of the following correctly defines Big-O notation?
   1) - A way to measure execution time exactly
   2) + A function that describes worst-case growth rate
   3) - A method for approximating runtime errors
   4) - A notation used only for best-case analysis

11) What is the asymptotic complexity of a polynomial function of degree k?
- 1) - O(n)
- 2) - O(n^2)
- 3) - O(n^3)
- 4) + O(n^k)

12) Which of the following is TRUE. about Theta (Θ) notation?
- 1) - It provides only an upper bound on performance
- 2) + It describes the tight bound of an algorithm's growth rate
- 3) - It is only used for best-case analysis
- 4) - It ignores the highest-order term

13) Insertion Sort is an in-place sorting algorithm.
- 1) + TRUE.
- 2) - FALSE.

14) The best-case runtime complexity of Insertion Sort occurs when the input is already sorted.
- 1) + TRUE.
- 2) - FALSE.

15) What is the primary advantage of Merge Sort over Insertion Sort?
- 1) + It has a lower worst-case time complexity
- 2) - It requires no additional memory
- 3) - It is easier to implement
- 4) - It sorts data in-place

16) What is the worst-case time complexity of Merge Sort?
- 1) - O(n)
- 2) + O(n log n)
- 3) - O(n^2)
- 4) - O(log n)

17) Merge Sort follows which of the following paradigms?
- 1) - Greedy
- 2) - Dynamic Programming
- 3) + Divide and Conquer
- 4) - Backtracking

18) What does the Merge function in Merge Sort do?
- 1) - Splits the array into two halves
- 2) + Merges two sorted subarrays into a single sorted array
- 3) - Sorts individual elements in an array
- 4) - Removes duplicate values from an array

19) The recurrence relation for Merge Sort is given by:
- 1) - T(n) = T(n-1) + O(n)
- 2) + T(n) = 2T(n/2) + O(n)
- 3) - T(n) = T(n-1) + O(log n)
- 4) - T(n) = O(n^2)

20) Which of the following methods can be used to solve recurrences?
- 1) - Substitution method
- 2) - Iteration method
- 3) - Master theorem
- 4) + All of the mentioned

21) Which recurrence solving technique involves making an educated guess and proving it via induction
- 1) - Iteration method
- 2) - Master theorem

3) + Substitution method
4) - Recursion tree method

22) The Master Theorem is used to solve recurrence relations of the form:
1) + T(n) = aT(n/b) + O(n^d)
2) - T(n) = aT(n-1) + O(n)
3) - T(n) = O(n²)
4) - T(n) = nT(n/2) + O(log n)

23) If a recurrence relation follows T(n) = 4T(n/2) + O(n²), what is the time complexity using the Master Theorem?
1) - O(n log n)
2) - O(n²)
3) - O(n³)
4) + All mentioned are FALSE.

24) Merge Sort is an in-place sorting algorithm.
1) - TRUE.
2) + FALSE.

25) The best-case time complexity of Merge Sort is O(n log n).
1) + TRUE.
2) - FALSE.

26) What is the purpose of the Master Theorem?
1) - To solve linear recurrence relations
2) - To determine the exact execution time of an algorithm
3) + To provide asymptotic bounds for divide-and-conquer recurrences
4) - To compare the space complexity of different algorithms

27) Which of the following is NOT a method for solving recurrence relations?
1) - Iteration method
2) - Substitution method
3) + Integration method
4) - Master theorem

28) If a = b^d in the recurrence T(n) = aT(n/b) + O(n^d), what is the asymptotic complexity?
1) - O(n log n)
2) + O(n^d log n)
3) - O(n^d)
4) - O(log n)

29) How is a binary heap represented in an array?
1) + The root is at index 1, and children of node i are at 2i and 2i+1
2) - The root is at index 0, and children of node i are at i+1 and i+2
3) - Nodes are stored in a linked list format
4) - Using a hash table

30) What is the purpose of the Heapify() function?
1) - To create a new heap from an unsorted array
2) - To insert a new element into a heap
3) + To maintain the heap property after modifications
4) - To remove the maximum element from a heap

31) Which operation has the highest time complexity in Heapsort?
1) + Building the heap
2) - Extracting the maximum element
3) - Inserting an element
4) - Heapify()

32) Which of the following sorting algorithms has the same worst-case time complexity as Heapsort?
1) - QuickSort
2) - Bubble Sort
3) + Merge Sort
4) - Selection Sort

33) The BuildHeap() function constructs a heap in O(n log n) time complexity.
1) - TRUE.
2) + FALSE.

34) The height of a heap with n elements is O(log n).
1) + TRUE.
2) - FALSE.

35) What is the worst-case time complexity of Quicksort?
1) - O(n)
2) - O(n log n)
3) + $O(n^2)$
4) - O(log n)

36) Which of the following best describes the average-case time complexity of Quicksort?
1) - O(n)
2) + O(n log n)
3) - $O(n^2)$
4) - O(log n)

37) Which of the following statements about the heap property is TRUE.?
1) - The parent node is always smaller than its child nodes
2) + A min-heap stores the smallest value at the root
3) - The largest value in a max-heap is always stored in a leaf node
4) - A heap must be a full binary tree

38) What is the time complexity of the Heapify() function?
1) - O(1)
2) - O(n)
3) + O(log n)
4) - O(n log n)

39) In a binary heap stored as an array, where is the left child of a node at index i located?
1) - A[i/2]
2) - A[i+1]
3) + A[2i]
4) - A[2i+1]

40) What does the Partition() function do in Quicksort?
1) - Splits the array into two equal halves
2) + Divides the array around a pivot element such that elements on one side are smaller and on the other are larger
3) - Merges two sorted subarrays
4) - Finds the median of the array

41) What is the worst-case time complexity of BuildHeap()?
1) + O(n)
2) - O(n log n)
3) - O(log n)
4) - $O(n^2)$

42) Heapsort is an in-place sorting algorithm.
1) + TRUE.

2) - FALSE.

43) The Partition() function in Quicksort always selects the median as the pivot.
1) - TRUE.
2) + FALSE.

44) Which technique does Quicksort use to divide the array?
1) - Merging
2) + Partitioning around a pivot
3) - Finding the median
4) - Insertion at correct positions

45) What happens in the partitioning step of Quicksort?
1) - The array is split into exactly equal halves
2) - The largest element is placed at the end
3) + Elements smaller than the pivot go to the left, larger ones go to the right
4) - The array is sorted in one step

46) In Quicksort, when does the worst case occur?
1) - When the pivot is always the median
2) + When the pivot is always the smallest or largest element
3) - When the array is already sorted
4) - When the input is random

47) Which of the following modifications can improve Quicksort's performance on sorted data?
1) - Using the first element as the pivot
2) - Always selecting the last element as the pivot
3) + Choosing a random pivot
4) - Using Bubble Sort instead

48) Which of the following is a key difference between Quicksort and Merge Sort?
1) - Quicksort has a worst-case of O(n log n) while Merge Sort is $O(n^2)$
2) + Quicksort sorts in-place while Merge Sort requires additional space
3) - Quicksort is a stable sort while Merge Sort is not
4) - Quicksort always runs faster than Merge Sort

49) Quicksort is an example of a divide-and-conquer algorithm.
1) + TRUE.
2) - FALSE.

50) The best-case scenario for Quicksort happens when the pivot always divides the array into equal halves.
1) + TRUE.
2) - FALSE.

51) Using a random pivot selection in Quicksort helps prevent the worst-case time complexity of $O(n^2)$.
1) + TRUE.
2) - FALSE.

52) The partition step in Quicksort requires $O(n^2)$ time.
1) - TRUE.
2) + FALSE.

53) The average-case time complexity of Quicksort is worse than Merge Sort.
1) - TRUE.
2) + FALSE.

54) What is the main drawback of comparison-based sorting algorithms?
1) - They require additional memory
2) + Their best possible time complexity is O(n log n)
3) - They are not stable
4) - They cannot be implemented in-place

55) Which of the following sorting algorithms is NOT a comparison sort?
1) - Merge Sort
2) - Quick Sort
3) + Counting Sort
4) - Heap Sort

56) Counting Sort works best when:
1) - The range of input values is much larger than the number of elements
2) + The range of input values is small and known in advance
3) - The input consists of floating-point numbers
4) - The input is already sorted

57) What is the time complexity of Counting Sort?
1) - $O(n^2)$
2) - $O(n \log n)$
3) + $O(n + k)$
4) - $O(k \log n)$

58) Which of the following statements about Counting Sort is TRUE.?
1) - It is an in-place sorting algorithm
2) + It is a stable sorting algorithm
3) - It is a comparison-based sorting algorithm
4) - It has a worst-case time complexity of $O(n^2)$

59) Which sorting algorithm is best suited for sorting large numbers with a fixed number of digits?
1) - Quick Sort
2) - Merge Sort
3) + Radix Sort
4) - Heap Sort

60) What is the key idea behind Radix Sort?
1) - Sorting the numbers digit by digit starting from the most significant digit
2) - Sorting numbers by dividing them into smaller groups
3) + Sorting numbers digit by digit starting from the least significant digit
4) - Using comparisons to determine the correct order

61) Which sorting technique is often used as a subroutine in Radix Sort?
1) - Merge Sort
2) - Quick Sort
3) + Counting Sort
4) - Heap Sort

62) Radix Sort can be used to sort floating-point numbers.
1) + TRUE.
2) - FALSE.

63) Counting Sort always requires additional space proportional to the range of input values.
1) + TRUE.
2) - FALSE.

64) What is the worst-case time complexity of Bucket Sort when input is uniformly distributed?
1) + O(n)
2) - O(n log n)
3) - $O(n^2)$
4) - O(log n)

65) What is the worst-case time complexity of finding the minimum element in an unsorted array of size n?
1) - O(1)
2) - O(log n)

3) + O(n)
4) - O(n log n)

66) Which technique does the Randomized Selection algorithm use to find the ith smallest element?
1) - Merge Sort
2) - Heap Sort
3) + Partitioning like Quick Sort
4) - Using a Binary Search Tree

67) What is the expected time complexity of Randomized Selection?
1) - $O(n^2)$
2) - O(n log n)
3) + O(n)
4) - O(log n)

68) Radix Sort is an in-place sorting algorithm.
1) - TRUE.
2) + FALSE.

69) Bucket Sort is efficient when input elements are uniformly distributed.
1) + TRUE.
2) - FALSE.

70) Finding the median of an unsorted array can be done in O(n) worst-case time.
1) + TRUE.
2) - FALSE.

71) Which of the following operations is NOT typically supported by a dynamic set?
1) - Search(S, k)
2) - Minimum(S)
3) + Sort(S)
4) - Insert(S, x)

72) What is the key property of a Binary Search Tree (BST)?
1) - The root node contains the largest key
2) - Each node has at most one child
3) + Left subtree contains keys smaller than the node, and the right subtree contains keys greater than the node
4) - It is a complete binary tree

73) What is the time complexity of searching for a key in a BST in the worst case?
1) - O(1)
2) - O(log n)
3) + O(n)
4) - O(n log n)

74) In an inorder traversal of a BST, how are the elements printed?
1) - In decreasing order
2) - In random order
3) + In sorted order
4) - Only the leaf nodes are printed

75) What is the best-case time complexity of inserting an element into a BST?
1) - O(1)
2) + O(log n)
3) - O(n)
4) - O(n log n)

76) Which of the following cases will cause the worst-case height of a BST to be O(n)?
1) - The tree is perfectly balanced

2) - The tree is built randomly
3) + The tree degenerates into a linked list
4) - The tree contains duplicate elements

77) What is the successor of a node x in a BST if it has right subtree?
1) - The smallest node in the left subtree of x
2) - The largest node in the right subtree of x
3) + The smallest node in the right subtree of x
4) - The parent of x

78) What is the time complexity of an inorder tree traversal in a BST?
1) - O(1)
2) - O(log n)
3) + O(n)
4) - O(n log n)

79) The worst-case height of a BST is O(log n).
1) - TRUE.
2) + FALSE.

80) If a BST is balanced, search operations will run in O(log n) time.
1) + TRUE.
2) - FALSE.

81) Which sorting algorithm is NOT an in-place sorting algorithm?
1) - QuickSort
2) + Merge Sort
3) - Heap Sort
4) - Insertion Sort

82) In the Master Theorem, if T(n) = aT(n/b) + O(n^d), and d > log_b(a), what is the time complexity?
1) - O(n log n)
2) + O(n^d)
3) - O(n^log_b(a))
4) - O(log n)

83) What is the primary idea behind Dynamic Programming?
1) - Breaking a problem into unrelated subproblems
2) - Solving problems by recursion without storing results
3) + Storing solutions to overlapping subproblems to avoid redundant computation
4) - Using a brute-force approach to explore all possibilities

84) Which of the following problems is best solved using Dynamic Programming?
1) - Binary Search
2) + Fibonacci Sequence Calculation
3) - Bubble Sort
4) - Depth-First Search

85) In the top-down approach to Dynamic Programming, which technique is commonly used to avoid redundant calculations?
1) - Iteration
2) + Memoization
3) - Backtracking
4) - Greedy algorithms

86) The top-down approach in Dynamic Programming is also called memoization.
1) + TRUE.
2) - FALSE.