Algorithmics MCQs

Hashing

Separate Chaining, Linear Probing, Quadratic Probing

1. In a hash table of size 11 using separate chaining, what is the index where the key 315 would be stored if the hash function is $h(\text{key}) = \text{key}\%11$?

a) 0

b) 3

c) 7

d) 11

2. Using linear probing with a hash function $h(\text{key}) = \text{key}\%10$, what would be the sequence of probed indices when inserting 25, 35, 45, 55 into an initially empty hash table of size 10?

a) 5, 5, 5, 5

b) 5, 5, 6, 7

c) 5, 5, 5, 6

d) 5, 6, 7, 8

3. In quadratic probing, if the initial probe position is $h(k)$, what is the i-th probe position?

a) $h(k) + i$

b) $h(k) + i^2$

c) $h(k) \cdot i$

d) $h(k) + i + i^2$

4. Which of the following is NOT an advantage of separate chaining over open addressing?

a) Simpler to implement

b) Better for high load factors

c) Supports more efficient deletion

d) Uses less memory

5. In a hash table using linear probing, what is the primary clustering problem?

a) Keys with different hash values cluster together

b) Keys with the same hash value form separate clusters

c) All keys cluster at the beginning of the table

d) All keys cluster at the end of the table

6. What is the time complexity of insertion in a hash table using separate chaining in the average case?

a) $O(1)$

b) $O(\log n)$

c) $O(n)$

d) $O(n^2)$

7. In quadratic probing, what happens if all positions given by the probing sequence are occupied?

a) The key is not inserted

b) The table size is doubled

c) Linear probing is used

d) The key replaces an existing key

8. Which hashing technique is most likely to suffer from secondary clustering?

a) Separate chaining

b) Linear probing

c) Quadratic probing

d) Double hashing

9. What is the load factor of a hash table?

a) The number of keys divided by the table size

b) The table size divided by the number of keys

c) The number of collisions divided by the table size

d) The number of empty slots divided by the table size

10. In a hash table of size 10 using quadratic probing, if a key hashes to index 3, what would be the sequence of probed indices?

a) 3, 4, 5, 6, 7

b) 3, 4, 6, 9, 3

c) 3, 4, 7, 2, 9

d) 3, 5, 9, 5, 3

Tree Data Structures

Binary Search Trees (BST), Min-Heaps, AVL Trees, B-Trees

11. In a binary search tree, where is the smallest element located?

a) Root

b) Leftmost leaf

c) Rightmost leaf

d) Any leaf

12. What is the time complexity of inserting an element into a balanced AVL tree?

a) $O(1)$

b) $O(\log n)$

c) $O(n)$

d) $O(n \log n)$

13. In a min-heap, what is true about the root node?

a) It's the largest element

b) It's the smallest element

c) It's always a leaf

d) It has no children

14. What is the maximum number of rotations needed to balance an AVL tree after a single insertion?

a) 1

b) 2

c) 3

d) $\log n$

15. In a B-tree of order m, what is the minimum number of keys in a non-root internal node?

a) $m/2$

b) $m - 1$

c) $\lceil m/2 \rceil - 1$

d) m

16. Which tree structure is most efficient for range queries?

a) Binary Search Tree

b) AVL Tree

c) B-Tree

d) Min-Heap

17. What is the height of a perfectly balanced binary tree with n nodes?

a) $\log n$

b) $n/2$

c) $\sqrt{n}$

d) n

18. In an AVL tree, what is the maximum allowed difference between the heights of left and right subtrees?

a) 0

b) 1

c) 2

d) $\log n$

19. What is the time complexity of finding the maximum element in a binary search tree?

a) $O(1)$

b) $O(\log n)$

c) $O(n)$

d) $O(h)$, where h is the height of the tree

20. In a B-tree, what happens when a node becomes overfull during insertion?

a) The node splits

b) A new level is added to the tree

c) The tree is rebalanced

d) The insertion is rejected

Sorting Algorithms

Quick Sort, Merge Sort, Insertion Sort

21. What is the average-case time complexity of QuickSort?

a) $O(n)$

b) $O(n \log n)$

c) $O(n^2)$

d) $O(\log n)$

22. Which sorting algorithm is most efficient for nearly sorted data?

a) QuickSort

b) MergeSort

c) InsertionSort

d) HeapSort

23. What is the space complexity of MergeSort?

a) $O(1)$

b) $O(\log n)$

c) $O(n)$

d) $O(n \log n)$

24. In QuickSort, what is the ideal choice for a pivot to ensure best performance?

a) First element

b) Last element

c) Middle element

d) Median of the array

25. Which sorting algorithm is guaranteed to have $O(n \log n)$ time complexity in all cases?

a) QuickSort

b) MergeSort

c) InsertionSort

d) BubbleSort

26. What is the best-case time complexity of InsertionSort?

a) $O(1)$

b) $O(n)$

c) $O(n \log n)$

d) $O(n^2)$

27. Which sorting algorithm is most suitable for sorting linked lists?

a) QuickSort

b) MergeSort

c) HeapSort

d) RadixSort

28. In the worst case, how many comparisons does QuickSort make?

a) $n \log n$

b) n^2

c) $n^2/2$

d) 2^n

29. What is the main advantage of MergeSort over QuickSort?

a) Better average-case performance

b) In-place sorting

c) Stable sorting

d) Simpler implementation

30. Which sorting algorithm is used by Java for sorting object arrays?

a) QuickSort

b) MergeSort

c) HeapSort

d) TimSort

Graph Algorithms

Minimum Spanning Trees (Kruskal's and Prim's algorithms), Shortest Path Algorithms (Dijkstra's algorithm), Eulerian Graphs, Hamiltonian Cycles

31. Which algorithm is used to find the minimum spanning tree of a graph?

a) Dijkstra's

b) Bellman-Ford

c) Kruskal's

d) Floyd-Warshall

32. In Prim's algorithm, how is the next vertex chosen to add to the minimum spanning tree?

a) Lowest degree

b) Highest degree

c) Lowest weight edge connecting to the tree

d) Random selection

33. What is the time complexity of Dijkstra's algorithm using a binary heap?

a) $O(V^2)$

b) $O(E \log V)$

c) $O(V + E)$

d) $O(V \log V)$

34. Which of the following is a characteristic of an Eulerian circuit?

a) Visits every vertex once

b) Visits every edge once

c) Starts and ends at different vertices

d) Contains no cycles

35. What is a necessary condition for a graph to have a Hamiltonian cycle?

a) The graph must be complete

b) The graph must be bipartite

c) Every vertex must have degree at least 2

d) The graph must be planar

36. In Kruskal's algorithm, what data structure is commonly used to detect cycles?

a) Stack

b) Queue

c) Disjoint Set

d) Binary Heap

37. What is the time complexity of finding an Eulerian circuit in a graph?

a) $O(V)$

b) $O(E)$

c) $O(V + E)$

d) $O(V \log E)$

38. Which algorithm is most suitable for finding shortest paths in a graph with negative edge weights?

a) Dijkstra's

b) Bellman-Ford

c) Floyd-Warshall

d) A*

39. What is the maximum number of edges in a minimum spanning tree of a graph with n vertices?

a) n

b) $n - 1$

c) $n(n-1)/2$

d) $2n-1$

40. In a complete graph with n vertices, how many Hamiltonian cycles are there?

a) n

b) $n!$

c) $(n-1)!$

d) n^n

Dynamic Programming

Forward and Backward Algorithms, Minimum Cost Paths

41. What is the primary advantage of using dynamic programming?

a) Reduced time complexity

b) Reduced space complexity

c) Simplicity of implementation

d) Guaranteed optimal solution

42. In the context of dynamic programming, what does "optimal substructure" mean?

a) The problem can be solved using greedy algorithms

b) The optimal solution can be constructed from optimal solutions of subproblems

c) The problem has no polynomial-time solution

d) The problem requires exponential time to solve

43. Which of the following problems is typically solved using dynamic programming?

a) Finding the shortest path in an unweighted graph

b) Sorting an array

c) The knapsack problem

d) Finding a minimum spanning tree

44. What is the time complexity of the Floyd-Warshall algorithm for all-pairs shortest paths?

a) $O(V^2)$

b) $O(V^3)$

c) $O(VE)$

d) $O(E \log V)$

45. In the context of the rod cutting problem, what does the optimal substructure property imply?

a) The problem can be solved using greedy algorithms

b) The optimal solution for a rod of length n includes optimal solutions for smaller lengths

c) The problem has no polynomial-time solution

d) The problem requires exponential time to solve

46. What is memoization in the context of dynamic programming?

a) A technique to reduce memory usage

b) A method to store results of expensive function calls

c) A way to optimize recursive algorithms

d) A process to convert recursive algorithms to iterative ones

47. Which approach does the forward algorithm in dynamic programming typically use?

a) Top-down

b) Bottom-up

c) Greedy

d) Divide and conquer

48. In the context of the minimum cost path problem, what does the backward algorithm typically do?

a) Starts from the end and moves towards the start

b) Starts from the start and moves towards the end

c) Randomly selects paths until the optimal one is found

d) Uses a greedy approach to select the next step

49. What is the space complexity of the standard dynamic programming solution to the 0/1 Knapsack problem?

a) $O(n)$

b) $O(W)$

c) $O(nW)$

d) $O(2^n)$

50. In solving the longest common subsequence problem using dynamic programming, what is typically stored in the DP table?

a) The characters of the subsequence

b) The length of the longest common subsequence

c) The indices of the matching characters

d) The entire subsequence at each step

Mathematical Proofs and Theorems

Degree of Vertices in Graphs, Maximum Matching in Graphs

51. What is the sum of all vertex degrees in an undirected graph with E edges?

a) E

b) $2E$

c) $E/2$

d) E^2

52. In a simple graph, what is the maximum number of edges possible with n vertices?

a) n

b) $n - 1$

c) $n(n - 1)/2$

d) 2^n

53. What is the minimum number of vertices a graph must have to guarantee that it contains either a clique of size 3 or an independent set of size 3?

a) 3

b) 5

c) 6

d) 9

54. In a bipartite graph, what is true about the sum of degrees of vertices in each partition?

a) They are always equal

b) They are always different

c) The sum in one partition is always greater

d) There is no relation between the sums

55. What is the maximum size of a matching in a bipartite graph with n vertices on each side?

a) n

b) $n - 1$

c) $2n$

d) n^2

56. In a graph with n vertices, what is the minimum number of edges required to ensure the graph is connected?

a) n

b) $n - 1$

c) $n(n - 1)/2$

d) $\log n$

57. What does the handshaking lemma state about an undirected graph?

a) The number of odd-degree vertices is even

b) The number of even-degree vertices is odd

c) The sum of all degrees is even

d) The product of all degrees is even

58. In a planar graph, what relation holds between the number of vertices (V), edges (E), and faces (F)?

a) $V + E = F$

b) $V - E + F = 2$

c) $V + F = E$

d) $V * F = E$

59. What is the chromatic number of a complete graph with n vertices?

a) 1

b) 2

c) n

d) $n - 1$

60. In a tree with n vertices, how many edges are there?

a) n

b) $n - 1$

c) $n + 1$

d) $\log n$

Algorithm Analysis

Recurrence Relations, Time Complexity, Space Complexity

61. What is the time complexity of binary search on a sorted array of n elements?

a) $O(1)$

b) $O(\log n)$

c) $O(n)$

d) $O(n \log n)$

62. Which of the following recurrence relations represents the time complexity of Merge Sort?

a) $T(n) = 2T(n/2) + n$

b) $T(n) = T(n - 1) + 1$

c) $T(n) = 2T(n/2) + 1$

d) $T(n) = T(n/2) + n$

63. What is the space complexity of QuickSort in the worst case?

a) $O(1)$

b) $O(\log n)$

c) $O(n)$

d) $O(n^2)$

64. Which of the following is NOT a method for solving recurrence relations?

a) Master Theorem

b) Recursion Tree

c) Substitution Method

d) Bubble Sort

65. What is the time complexity of the best algorithm for matrix multiplication of two $n \times n$ matrices?

a) $O(n^2)$

b) $O(n^{2.37})$

c) $O(n^3)$

d) $O(2^n)$

66. In Big O notation, which of the following is correct?

a) $O(n^2) \subset O(n)$

b) $O(n) \subset O(n^2)$

c) $O(n) = O(n^2)$

d) $O(n)$ and $O(n^2)$ are incomparable

67. What is the time complexity of finding the kth smallest element in an unsorted array using QuickSelect algorithm on average?

a) $O(n)$

b) $O(n \log n)$

c) $O(k \log n)$

d) $O(n^2)$

68. Which of the following algorithms has a space complexity of $O(1)$?

a) MergeSort

b) QuickSort (in-place version)

c) HeapSort

d) CountingSort

69. What is the time complexity of the Floyd-Warshall algorithm for finding all-pairs shortest paths?

a) $O(V^2)$

b) $O(V^3)$

c) $O(VE)$

d) $O(E \log V)$

70. In analyzing recursive algorithms, what does the term ”memoization” refer to?

a) Reducing the problem size

b) Storing results of expensive function calls

c) Dividing the problem into subproblems

d) Combining solutions of subproblems

Answer Key with Simple Explanations

1. c) 7 - The hash function is $315\%11 = 7$.
2. b) 5, 5, 6, 7 - Linear probing moves to the next available slot.
3. b) $h(k) + i^2$ - This is the definition of quadratic probing.
4. d) Uses less memory - Separate chaining typically uses more memory due to linked lists.
5. a) Keys with different hash values cluster together - This is the primary clustering problem in linear probing.
6. a) $O(1)$ - Average case for insertion in separate chaining is constant time.
7. a) The key is not inserted - If all slots are full in quadratic probing, insertion fails.
8. c) Quadratic probing - Secondary clustering is a problem in quadratic probing.
9. a) The number of keys divided by the table size - This is the definition of load factor.
10. b) 3, 4, 6, 9, 3 - Quadratic probing uses the formula $h(k) + i^2$.
11. b) Leftmost leaf - The smallest element in a BST is at the leftmost leaf.
12. b) $O(\log n)$ - AVL trees maintain balance, ensuring logarithmic time for operations.
13. b) It's the smallest element - In a min-heap, the root is always the smallest element.
14. b) 2 - At most two rotations are needed to balance an AVL tree after insertion.
15. c) $\lceil m/2 \rceil - 1$ - This is the minimum number of keys in a non-root B-tree node.
16. c) B-Tree - B-Trees are efficient for range queries due to their structure.
17. a) $\log n$ - The height of a perfectly balanced binary tree is logarithmic.
18. b) 1 - AVL trees allow a maximum height difference of 1 between subtrees.
19. d) $O(h)$, where h is the height of the tree - Finding the maximum requires traversing to the rightmost node.
20. a) The node splits - When a B-tree node is overfull, it splits to maintain the tree's properties.

21. b) $O(n \log n)$ - This is the average-case time complexity of QuickSort.
22. c) InsertionSort - InsertionSort performs well on nearly sorted data.
23. c) $O(n)$ - MergeSort requires additional space proportional to the input size.
24. d) Median of the array - Choosing the median as pivot ensures balanced partitions.
25. b) MergeSort - MergeSort guarantees $O(n \log n)$ time complexity in all cases.
26. b) $O(n)$ - The best-case for InsertionSort is when the array is already sorted.
27. b) MergeSort - MergeSort is efficient for linked lists as it doesn't require random access.
28. b) n^2 - The worst case for QuickSort occurs with poor pivot selection.
29. c) Stable sorting - MergeSort is stable, while QuickSort is not.
30. d) TimSort - Java uses TimSort, a hybrid of MergeSort and InsertionSort, for object arrays.
31. c) Kruskal's - Kruskal's algorithm is commonly used to find minimum spanning trees.
32. c) Lowest weight edge connecting to the tree - This is how Prim's algorithm selects the next vertex.
33. b) $O(E \log V)$ - This is the time complexity of Dijkstra's algorithm with a binary heap.
34. b) Visits every edge once - An Eulerian circuit visits every edge exactly once.
35. c) Every vertex must have degree at least 2 - This is a necessary (but not sufficient) condition for a Hamiltonian cycle.
36. c) Disjoint Set - Disjoint Set (Union-Find) is used in Kruskal's algorithm to detect cycles.
37. c) $O(V + E)$ - Finding an Eulerian circuit can be done in linear time.
38. b) Bellman-Ford - Bellman-Ford can handle negative edge weights, unlike Dijkstra's algorithm.
39. b) $n - 1$ - A tree with n vertices has exactly $n - 1$ edges.
40. c) $(n-1)!$ - This is the number of Hamiltonian cycles in a complete graph.

41. a) Reduced time complexity - Dynamic programming often reduces time complexity by avoiding redundant computations.

42. b) The optimal solution can be constructed from optimal solutions of subproblems - This is the definition of optimal substructure.

43. c) The knapsack problem - The knapsack problem is a classic dynamic programming problem.

44. b) $O(V^3)$ - Floyd-Warshall has cubic time complexity.

45. b) The optimal solution for a rod of length n includes optimal solutions for smaller lengths - This illustrates optimal substructure.

46. b) A method to store results of expensive function calls - Memoization stores results to avoid redundant computations.

47. b) Bottom-up - The forward algorithm typically uses a bottom-up approach.

48. a) Starts from the end and moves towards the start - The backward algorithm works in reverse.

49. c) $O(nW)$ - The standard DP solution to 0/1 Knapsack uses a table of size $n \times W$.

50. b) The length of the longest common subsequence - The DP table typically stores lengths, not the actual subsequences.

51. b) $2E$ - Each edge contributes 2 to the sum of degrees.

52. c) $n(n-1)/2$ - This is the number of edges in a complete graph.

53. c) 6 - This is a result from Ramsey theory.

54. a) They are always equal - In a bipartite graph, the sum of degrees in each partition is equal.

55. a) n - The maximum matching in a bipartite graph can't exceed the number of vertices on one side.

56. b) $n-1$ - A tree is the minimal connected graph, and it has $n-1$ edges.

57. c) The sum of all degrees is even - This is the handshaking lemma.

58. b) $V - E + F = 2$ - This is Euler's formula for planar graphs.

59. c) n - A complete graph requires n colors, one for each vertex.

60. b) $n-1$ - A tree with n vertices always has $n-1$ edges.

61. b) $O(\log n)$ - Binary search halves the search space in each step.

62. a) $T(n) = 2T(n/2) + n$ - This recurrence represents Merge Sort's divide-and-conquer strategy.

63. b) $O(\log n)$ - The worst-case space complexity of QuickSort is logarithmic due to the recursion stack.

64. d) Bubble Sort - Bubble Sort is a sorting algorithm, not a method for solving recurrence relations.

65. b) $O(n^{2.37})$ - This is the current best known upper bound (Coppersmith–Winograd algorithm).

66. b) $O(n) \subset O(n^2)$ - $O(n)$ is a subset of $O(n^2)$.

67. a) $O(n)$ - QuickSelect has linear average-case time complexity.

68. c) HeapSort - HeapSort has a space complexity of $O(1)$ as it sorts the array in-place.

69. b) $O(V^3)$ - Floyd-Warshall algorithm has a time complexity of $O(V^3)$ for V vertices.

70. b) Storing results of expensive function calls - Memoization is a technique used to store and reuse the results of expensive function calls.