

38 Computer Science Fundamentals interview questions to hire top candidates

Questions

1. Can you explain the difference between a stack and a queue? Provide an example of when you'd use each in a real-world application.
2. What is recursion, and how would you implement it to solve the factorial problem?
3. Describe the concept of Big O notation. How would you determine the time complexity of a simple for loop?
4. What is the difference between a binary tree and a binary search tree? When would you choose one over the other?
5. Explain the concept of hashing. How would you handle collisions in a hash table?
6. What is the difference between depth-first search and breadth-first search? In what scenarios would you prefer one over the other?
7. Can you explain the concept of dynamic programming? Provide an example of a problem that can be solved using this technique.
8. What is the difference between a process and a thread? How do they relate to concurrent programming?
9. Explain the concept of object-oriented programming. How does inheritance contribute to code reusability?
10. What is the difference between a compiler and an interpreter? How does this affect the execution of code?
11. Can you explain the concept of a linked list and its advantages over an array?
12. What is a database transaction, and why is it important?
13. Can you describe the difference between synchronous and asynchronous operations?
14. What is a deadlock in the context of operating systems, and how can it be prevented?
15. Explain the concept of load balancing and why it is important in distributed systems.
16. What is the importance of normalization in a database?
17. Can you explain what a RESTful API is and its main principles?
18. Can you explain what a graph is and provide a real-world example of its application?
19. What are the advantages and disadvantages of using a hash table over a simple array?
20. How would you implement a priority queue, and what are its use cases?
21. Can you describe the concept of a trie and where it might be beneficial to use one?
22. Explain the difference between a singly linked list and a doubly linked list. When would you prefer to use one over the other?
23. What is a circular queue and how does it differ from a regular queue?
24. How do you traverse a binary tree, and what are the different methods to do so?
25. Can you explain the concept of a set and how it differs from a list?
26. What is the purpose of an adjacency list in representing a graph, and how does it compare to an adjacency matrix?
27. Can you describe a situation where a skip list would be more advantageous than other data structures?
28. What are the trade-offs between using an array versus a linked list in terms of memory usage and access times?
29. How would you implement a balanced binary search tree, and why is balancing important?
30. Can you explain what a binary tree is and its primary use cases?
31. What is a graph, and how is it different from a tree?
32. How would you describe an array and its advantages?
33. Can you explain what a priority queue is and where it might be used?
34. What is a trie, and how is it used in real-world applications?
35. How would you explain a dynamic array, and what are its benefits over a standard array?
36. What is a deque, and what are its typical use cases?
37. Can you describe the concept of a balanced tree and why it is important?
38. What is a sparse matrix and how is it different from a dense matrix?