CS Fundamentals

Interview Questions

For Freshers

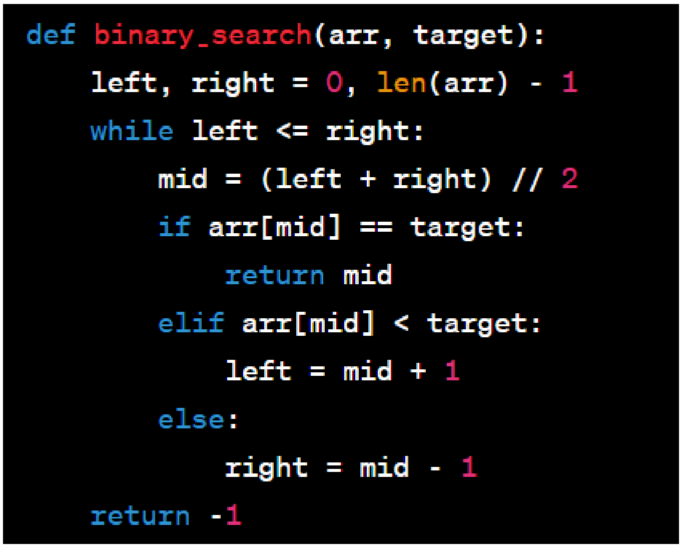
1. What is the difference between an array and a linked list?

- An array is a contiguous block of memory that stores elements of the same data type, accessed using an index. In contrast, a linked list is a data structure where elements, called nodes, are linked using pointers, and each node contains both the data and the address of the next node.

2. Explain the concept of time and space complexity in algorithms.

- Time complexity measures the amount of time an algorithm takes to run as a function of the input size. Space complexity measures the amount of memory space an algorithm uses as a function of the input size.

3. Implement a binary search algorithm for a sorted array.



```
def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

4. Define and compare the terms "stack" and "queue."

- A stack is a last-in, first-out (LIFO) data structure, where elements are added and removed from the same end (top). A queue is a first-in, first-out (FIFO) data structure, where elements are added at the rear and removed from the front.

5. What is recursion, and why is it useful in programming?

- Recursion is a programming technique where a function calls itself to solve smaller subproblems. It is useful for solving problems that can be broken down into similar subproblems, such as traversing trees or searching in sorted arrays.

6. Describe the process of traversing a binary tree in-order.

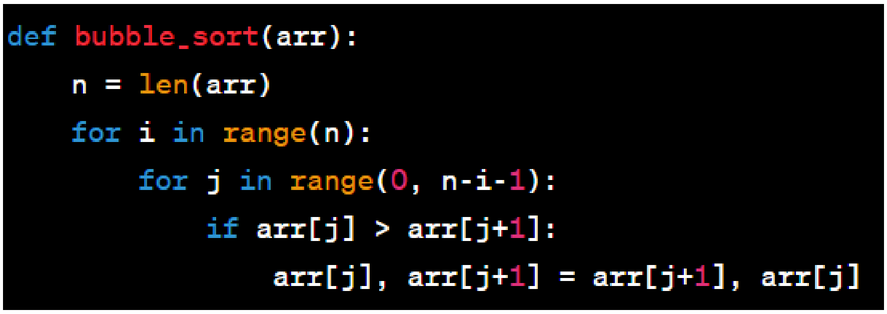
- In in-order traversal, the left subtree is visited first, then the root node, and finally the right subtree. This traversal results in elements being visited in ascending order in a binary search tree.

7. How does a hash table work, and what are its advantages?

- A hash table is a data structure that stores key-value pairs and uses a hash function to convert keys into array indices. This allows for efficient insertion, deletion, and retrieval of elements with average constant-time complexity.

8. Implement a basic sorting algorithm like bubble sort or insertion sort.

- Here's an example of bubble sort in Python:



```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
```

9. What is object-oriented programming (OOP), and why is it essential?

- OOP is a programming paradigm based on the concept of objects that encapsulate data and behavior. It helps organize code, promotes reusability, and enhances maintainability.

10. Explain the concepts of inheritance, encapsulation, and polymorphism.

- Inheritance allows a class (subclass) to inherit properties and methods from another class (superclass).

- Encapsulation is the concept of hiding internal implementation details of an object and providing access only through methods.

- Polymorphism allows objects to be treated as instances of their superclass, enabling multiple classes to be treated as instances of a common interface.

11. What is the difference between an abstract class and an interface?

- An abstract class may have both abstract and concrete methods, and it cannot be instantiated. An interface, on the other hand, only contains abstract method signatures and cannot have any method implementations. A class can implement multiple interfaces, but it can only inherit from one abstract class.

12. How do you handle exceptions in programming?

- Exceptions are handled using try-except blocks. Code that might raise an exception is placed inside the try block, and the exception is caught and handled in the except block.

13. Explain the principles of dynamic programming and provide an example.

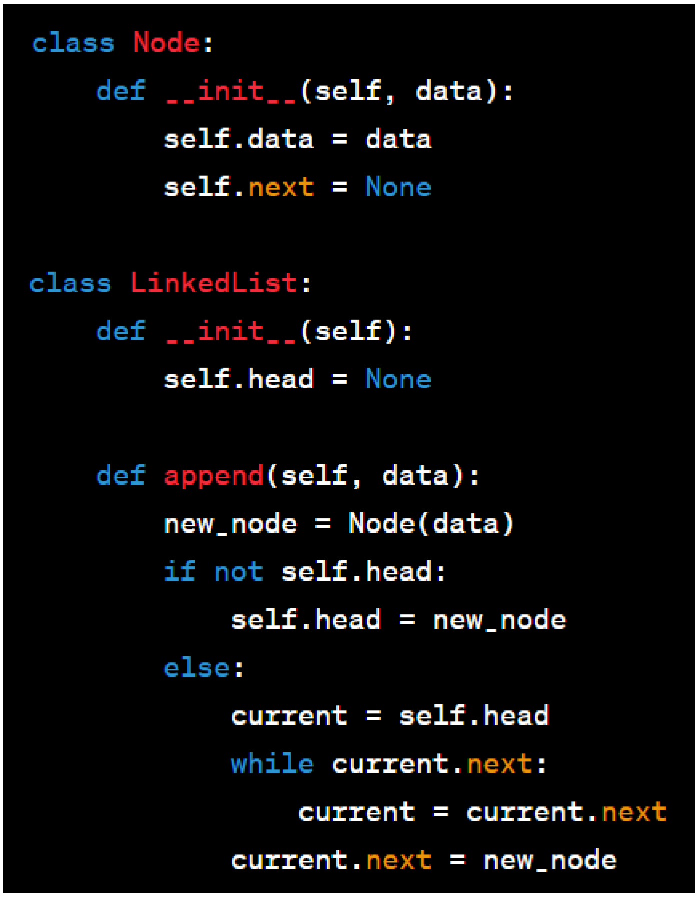
- Dynamic programming is an optimization technique that solves complex problems by breaking them down into overlapping subproblems and caching their solutions. Example: Fibonacci sequence calculation using memoization.

14. What is the difference between a breadth-first search (BFS) and a depth-first search (DFS)?

- BFS explores all neighbor nodes at the current depth before moving to the next level. DFS explores as far as possible along each branch before backtracking.

15. Implement a basic data structure like a linked list or a binary tree.

- Here's a simple implementation of a singly linked list in Python:



```python
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def append(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = new_node
        else:
            current = self.head
            while current.next:
                current = current.next
            current.next = new_node
```

16. Describe the concept of Big O notation and its significance.

- Big O notation is used to describe the upper bound of an algorithm's time or space complexity as the input size grows. It helps analyze and compare algorithm efficiency.

17. What are the key components of an operating system?

- Key components include the kernel, process management, memory management, file system, device drivers, and user interface.

18. Explain the difference between process and thread in the context of an operating system.

- A process is an independent unit of execution with its own memory space, whereas a thread is a lightweight unit of execution that shares the same memory space as other threads within the same process.

19. Discuss the basics of computer networking and the OSI model.

- The OSI model is a conceptual framework that standardizes the functions of a telecommunication or computing system into seven distinct layers, from physical transmission to application.

20. How does the internet work, and what are HTTP and HTTPS?

- The internet is a global network of interconnected devices that communicate using the TCP/IP protocol suite. HTTP (Hypertext Transfer Protocol) is used for transmitting data over the web, while HTTPS (HTTP Secure) encrypts the data for secure communication.

These answers should give you a solid foundation for your CS fundamentals interview preparation.

Make sure to study and practice further to gain a deeper understanding of these concepts and how they apply to real-world problem-solving.

Good luck!