DS & ALGOS SELF EVALUATION

CS Fundamentals (ie. data structures and algorithms) are the very first pillar in our 3 Pillars of Interviewing. Without having a good foundation of these, it's impossible to apply the strategies that you will learn throughout the rest of the course.

Trying to nail your coding interviews without mastering these fundamentals is like trying to win a game of basketball without knowing how to dribble the ball. All the strategy in the world isn't going to help you to win.

At this point in the course, we have to determine whether you should switch your attention to DS & Algos first or continue with learning the Coding Interview Success Framework. That's why we've created this evaluation.

The purpose of this evaluation is to give you a quick gauge of your current DS & Algos skill.

If you score 80% or above:

In this case, you may have more to learn about DS & Algos, but your current understanding is sufficient to move on to Module 2 of *Coding Interview Mastery*. Your time will be best spent by learning the Coding Interview Success Framework first and then using the material in this Primer simply to fill in the gaps.

If you score below 80%:

In this case, never fear. We've compiled all the resources in this Primer to help you dial in these skills. Take 1-2 weeks to work through the Primer and then revisit this assessment. Once you are able to score 80% or above, move on to Module 2 of *Coding Interview Mastery*.

QUESTIONS

1. What is the time complexity of the following code?

```
def f1(n: int):
    for i in range(n):
        for j in range(i+1, n):
            print(i+j)
```

Time Complexity: O(N^2)

2. What is the space complexity of the following code?

```
def f2(arr: List[int]):
    for i in range(len(arr)):
        arr[i] = arr[i]+1
```

Space Complexity: O(N)

3. Given an Array of length `N`, how many unique subarrays of length `K` does that array contain?

Answer: _____

4. True or False: Strings are immutable

Answer: True

5. A sliding window is used to:

A. Compare two strings for equality
B. Find substrings with certain characteristics
C. Make a copy of a strings
D. Identify prefixes and suffixes of a string

Answer: B

6. What is the time complexity of removing a node from a Linked List?

Answer: O(1)

7. What does the following code do?

```
def f3(head: ListNode) -> ListNode:
    prev = None
    curr = head

    while curr:
        next_temp = curr.next
        curr.next = prev
        prev = curr
        curr = next_temp

    return prev
```

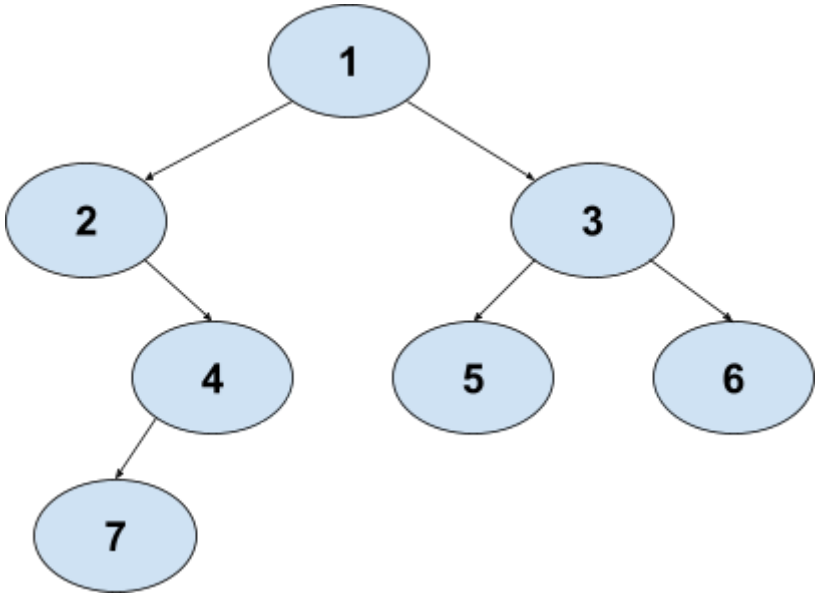
Answer:

Deletes the last node (or tail) of a Linked List.

8. True or False: All Binary Trees are Binary Search Trees

Answer: False

9. Given the following tree, what order would the nodes be visited in a Level Order Traversal?



Answer: 1->2->3->5->4->7->6

10. What are the two most common formats to represent a graph?

Answer: Vertexes or Indexes.

11. Why would you choose to use Breadth-first Search instead of Depth-first search?

Answer:

To solve.

12. What is the time complexity of inserting a value into a Heap?

Answer: _____

13. Heaps are generally implemented using what other data structure?

Answer: _____

14. LIFO and FIFO stand for what?

Answer: LIFO -< Last In First Out (STACK DS); FIFO -> First In First Out (QUEUE DS)

15. What is the time complexity of QuickSort?

Answer: _____

16. What property of an array must be true for you to use Binary Search?

Answer: _____

17. What is the output of the following code?

```
def f4(head: ListNode):
    if not head:
        return

    rem = f4(head.next)
    print(head.val)


f4(1 -> 2 -> 3 -> 4 -> null)
```

Answer: _____

18. What is the space complexity of the following code?

```
def f4(head: ListNode):
    if not head:
        return

    rem = f4(head.next)
    print(head.val)
```

Answer: _____

ANSWER KEY

To score this assessment, simply check your answers against the key. If you got less than 15 out of 18 answers correct, then I recommend working through the DS & Algos Primer first before moving on to Module 2 of *Coding Interview Mastery*.

1. What is the time complexity of the following code?

Answer: $O(n^2)$

2. What is the space complexity of the following code?

Answer: $O(1)$

3. Given an Array of length `N`, how many unique subarrays of length `K` does that array contain?

Answer: `N-K+1`

4. True or False: Strings are immutable

Answer: This depends on the programming language you use. Mark this as correct if you got the right answer for your preferred language

Java, C#, JavaScript, Python and Go: Immutable

Ruby, PHP, C, Swift (sometimes): Mutable

5. A sliding window is used to:

Answer: B. Find substrings with certain characteristics

6. What is the time complexity of removing a node from a Linked List?

Answer: $O(N)$

7. What does the following code do?

Answer: Reverse the Linked List in place and return the new head of the list.

8. True or False: All Binary Trees are Binary Search Trees

Answer: False

9. Given the following tree, what order would the nodes be visited in a Level Order Traversal?

Answer: `1, 2, 3, 4, 5, 6, 7`

10. What are the two most common formats to represent a graph?

Answer: Adjacency List and Adjacency Matrix

11. Why would you choose to use Breath-first Search instead of Depth-first search?

Answer: BFS will always find the shortest path first, so it is more efficient when we are attempting to find the shortest path. If we want to find the shortest path using DFS, we have to traverse every path, whereas with BFS we can stop as soon as we find a valid path.

12. What is the time complexity of inserting a value into a Heap?

Answer: `O(log N)`

13. Heaps are generally implemented using what other data structure?

Answer: Trees

14. LIFO and FIFO stand for what?

Answer: Last-in first-out, First-in first-out

15. What is the time complexity of QuickSort?

Answer: $O(N^2)$

Note: While the average complexity is $O(N \log N)$*, the worst case complexity is* $O(N^2)$

16. What property of an array must be true for you to use Binary Search?

Answer: The array must be sorted.

17. What is the output of the following code?

Answer:
4
3
2
1

18. What is the space complexity of the following code?

Answer: $O(N)$