# 7 DSA Approaches to get you hired

# Hashing

- Data Structure: Array and String
- Involves using a hash function to map data to an index in an array. Often used for fast lookups and storing key-value pairs.

# Divided and Conquer

- Data Structure: Array
- Usage: Involves breaking down a problem into smaller sub-problems that can be solved independently. Often used for sorting and searching algorithms.

# Slow and Fast Pointer
## (tortoise and hare)

- Data Structure: Linked List
- Usage: Involves using two pointers, one moving slowly and the other moving quickly through the linked list. This approach is particularly useful for detecting cycles in a linked list.

# Dynamic Programming

- Data Structure: Array and String
- Usage: Involves breaking a problem down into smaller sub-problems and solving each sub-problem only once. The results of each sub-problem are stored in memory to avoid duplicate calculations.

# Sliding Window

- Data Structure: Array
- Usage: Involves selecting a subarray of a given length from a larger array and then sliding it across the larger array to solve the problem.

# Depth First Search (DFS)

- Data Structure: Tree and Graph
- Usage: Involves exploring a tree or graph by going as deep as possible before backtracking. Often used for graph problems and maze-solving.

# Breadth First Search (BFS)

- Data Structure: Tree and Graph
- Usage: Involves exploring a tree or graph level by level. Used for finding the shortest path between two nodes in a graph.

You might need to learn a lot of things... but it doesn't have to be stressful