**Algorithm Strategy for improvement**

How to solve problems on Cracking the Coding Interview book (**CTCI, Leetcode, GeeksForGeeks, and more**. The single most important aspect of preparing for technical interviews is to solve many coding challenges. Although some tech companies have stopped asking algorithm questions and instead test domain-specific knowledge, most tech giants (and even the small ones) still require candidates to solve coding challenges. Many engineers get frustrated about this.

Getting frustrated is completely pointless and a waste of energy that can be spent on more productive ways.

The strategy:

1. Go to a quiet location.

2. Pick a question.

3. Set a timer for 25 minutes.

4. Allocate the first 5 minutes to understand the problem and ask/answer clarifying questions to yourself.

5. Spend the next 20 minutes solving the problem on paper & pen/pencil (until timer rings).

6. Spend the next 10 minutes studying the solution.

7. Stop. Even if you’re in the middle of something important like trying to understand the solution.

8. Do not spend more than 30–35 minutes per question.

9. Record activity on your progress chart.

10. Revisit the question next week.

Make sure to learn the following data structures before you watch this video:

* **Dynamic Array,**
* **Linked List,**
* **Stack & Queue,**
* **Hash Tables,**
* **Binary Search Tree,**
* **Binary Heaps & Priority,**
* **Queue,**
* **Graphs,**
* **Trie And review the following algorithms: Binary search, DFS (Depth-First Search),**
* **BFS (Breadth-First Search), Binary Tree Traversals (In-order, pre-order and post-order traversals), Traveling Salesman.**

No need to know: - Dijkstra Algorithm - Red-black trees - Other fancy data structures/algorithms.