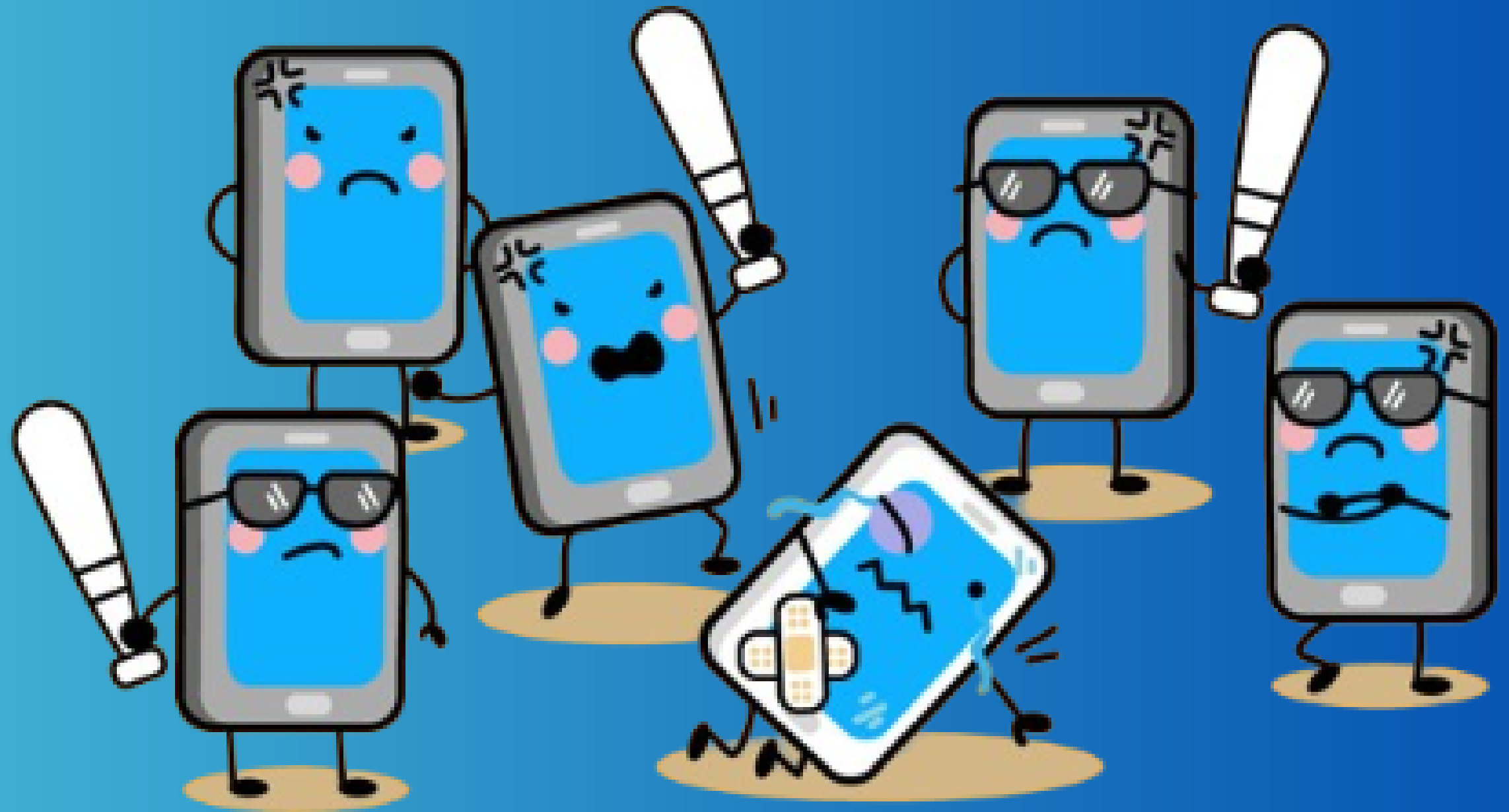
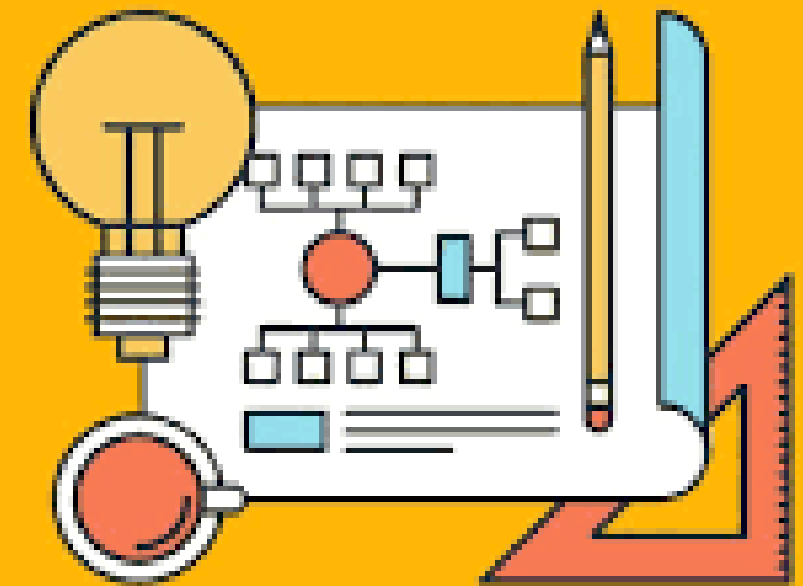


How to be good at Algorithms?



Data Structures and Algorithms

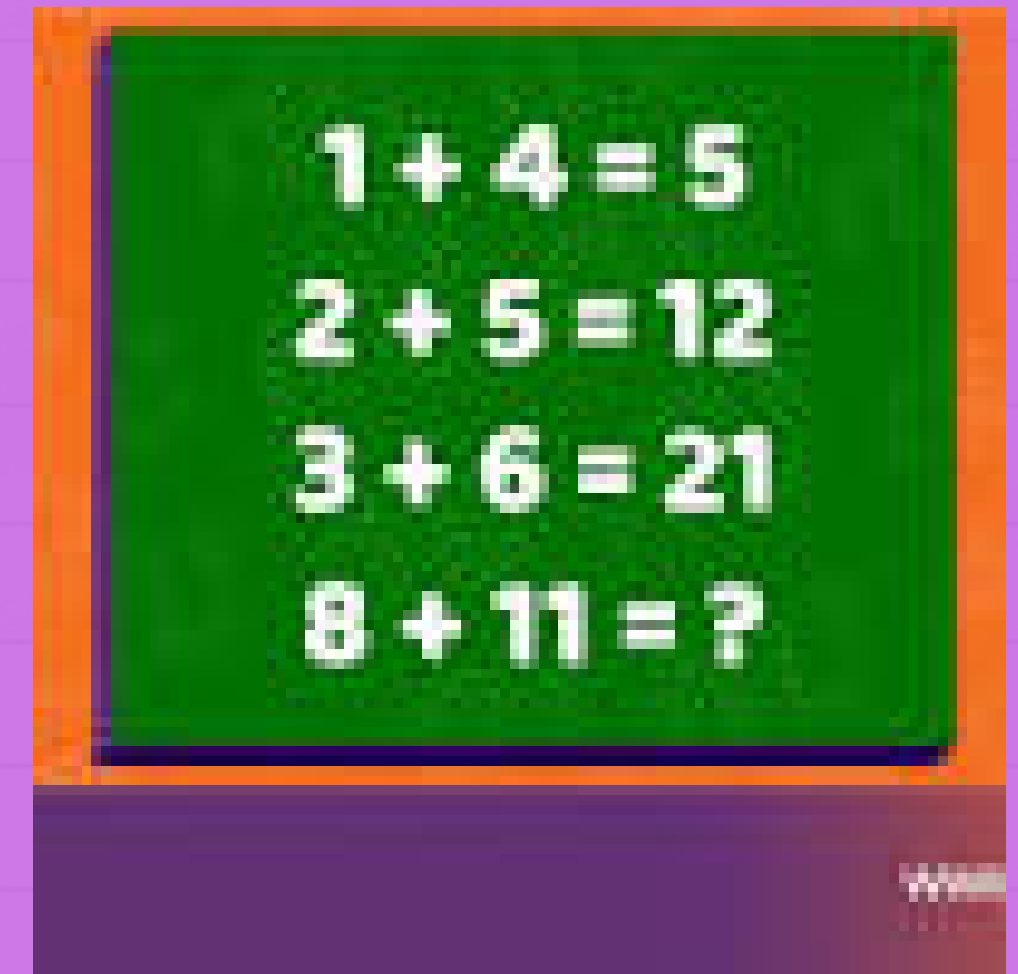
- This is absolutely essential, and your chosen methods for learning (online courses, textbooks, videos) are all great options. Choose resources that match your learning style and provide good explanations with practical examples.



**Data Structure and
Algorithm**

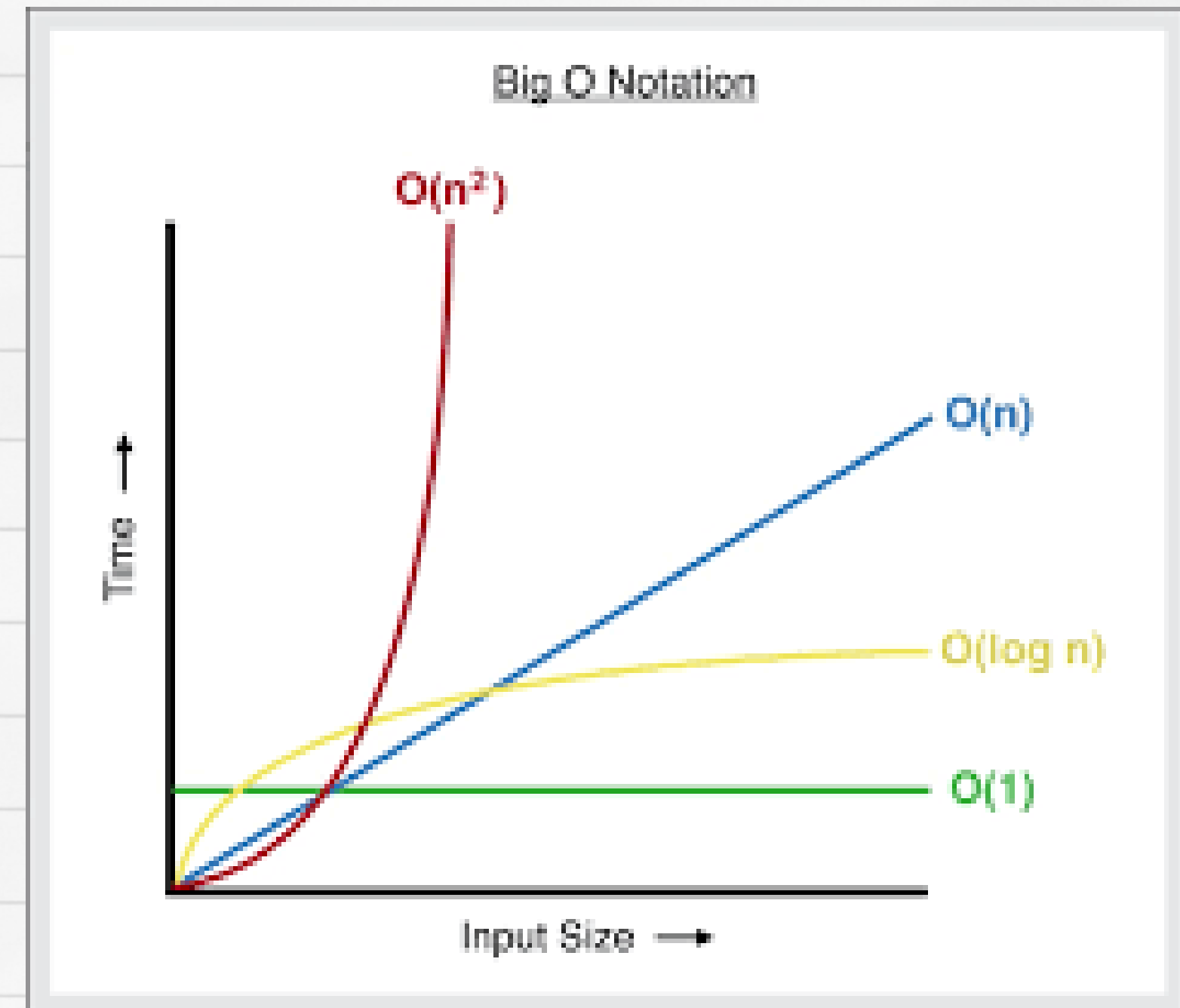
Math and Logic

- Sharpening your logic skills through riddles and puzzles is a fun and effective way to improve your problem-solving approach. Gradually increasing the difficulty will keep you challenged and engaged.



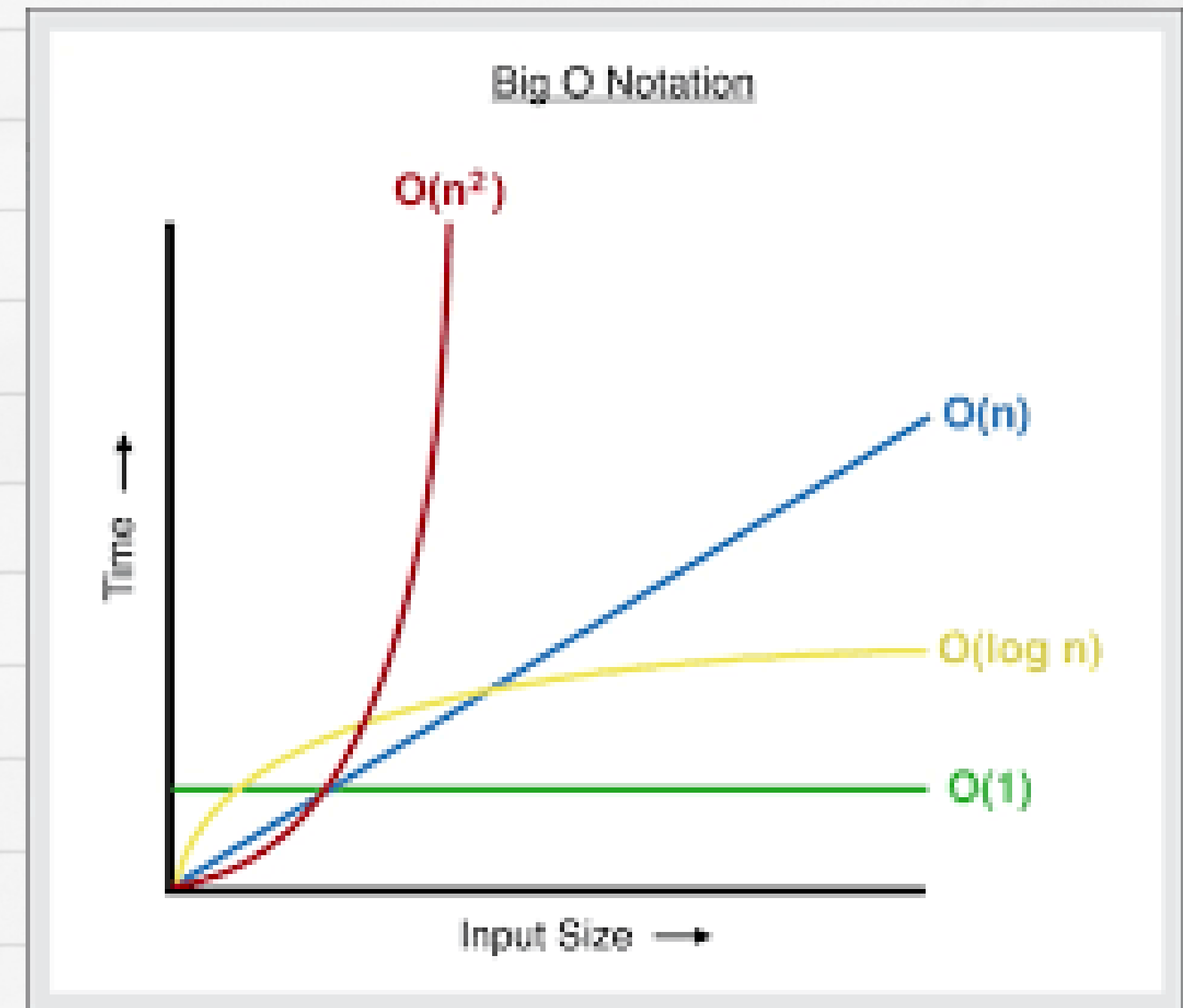
Big-O Notation

- Understanding how to analyze the time and space complexity of algorithms is crucial for choosing the most efficient solution. Make sure you can confidently analyze common algorithms and compare their performance.



Implementation

- Putting your theoretical knowledge into practice by implementing algorithms in your chosen programming language is vital. This will solidify your understanding and build practical skills.



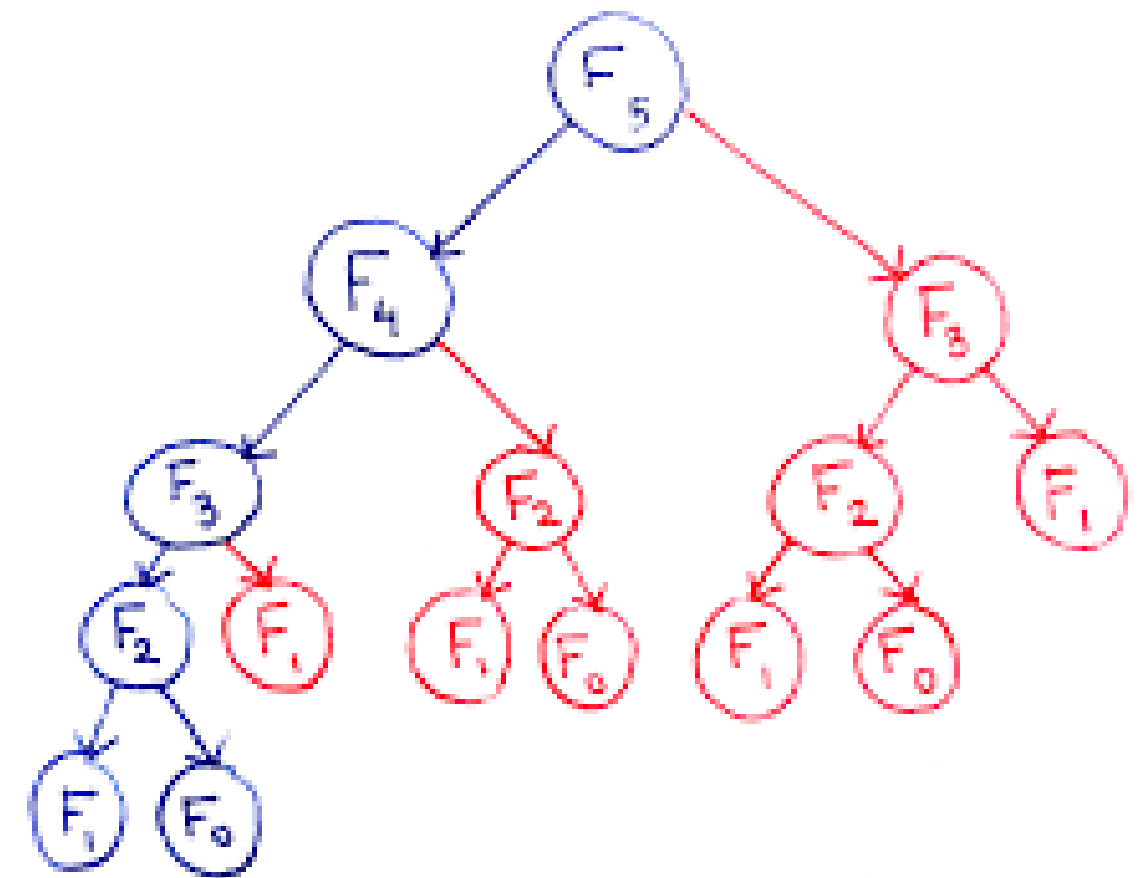
Challenge

- Participating in daily coding challenges is a fantastic way to stay motivated and improve your problem-solving skills under pressure. Platforms like LeetCode, HackerRank, and CodeForces offer a variety of problems at different difficulty levels.



Dynamic Programming

- This is a powerful optimization technique that you should definitely learn. It might seem advanced initially, but practicing problems will help you gain confidence and master its potential.



Interview Questions

- Familiarizing yourself with common interview questions will help you prepare for future coding interviews. Practice solving these problems under timed conditions to build your confidence and refine your problem-solving approach.



Additional tips

- Consistency is key
- Join the community
- Don't get discouraged





Many more ...



SWIPE