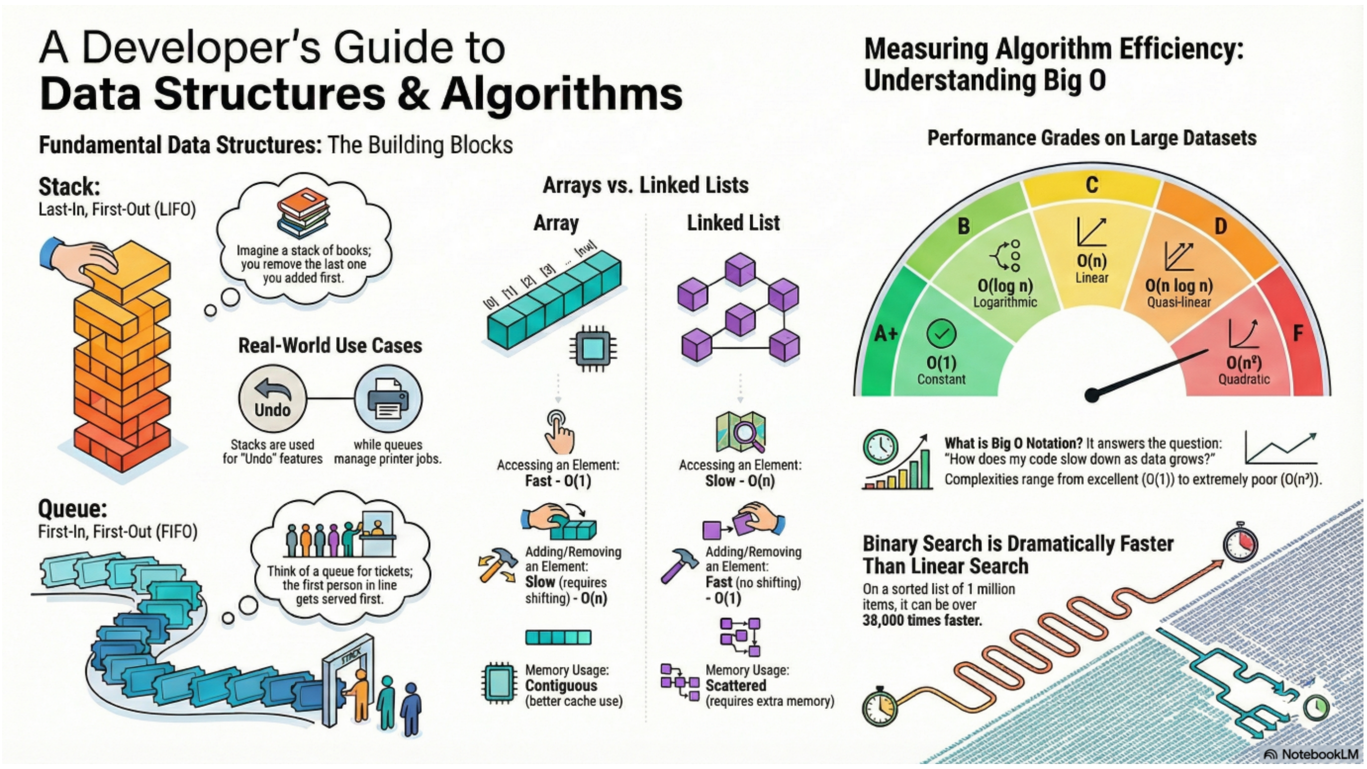
kintsugi-stack-dsa-cpp : COMPLEXITY

> “Talk is cheap. Show me the time complexity.”

- Author: Kintsugi-Programmer


A Developer's Guide to
Data Structures & Algorithms
Fundamental Data Structures: The Building Blocks
Stack:
Last-In, First-Out (LIFO)
Imagine a stack of books; you remove the last one you added first.
Real-World Use Cases
Undo
Stacks are used for "Undo" features
while queues manage printer jobs.
Queue:
First-In, First-Out (FIFO)
Think of a queue for tickets; the first person in line gets served first.
Arrays vs. Linked Lists
Array
Linked List
[0] [1] [2] [3] ... [n-1]
Accessing an Element: Fast - O(1)
Accessing an Element: Slow - O(n)
Adding/Removing an Element: Slow (requires shifting) - O(n)
Adding/Removing an Element: Fast (no shifting) - O(1)
Memory Usage: Contiguous (better cache use)
Memory Usage: Scattered (requires extra memory)
Measuring Algorithm Efficiency: Understanding Big O
Performance Grades on Large Datasets
A+ O(1) Constant
B O(log n) Logarithmic
C O(n) Linear
D O(n log n) Quasi-linear
F O(n²) Quadratic
What is Big O Notation? It answers the question: "How does my code slow down as data grows?" Complexities range from excellent (O(1)) to extremely poor (O(n²)).
Binary Search is Dramatically Faster Than Linear Search
On a sorted list of 1 million items, it can be over 38,000 times faster.
NotebookLM

> Disclaimer: The content presented here is a curated blend of my personal learning journey, experiences, open-source documentation, and invaluable knowledge gained from diverse sources. I do not claim sole ownership over all the material; this is a community-driven effort to learn, share, and grow together.

End-of-File