Week 1: Fundamentals of Programming

G	nal	le.
\mathbf{L}	con	· .

- Refresh programming basics: loops, functions, recursion, and complexity.
- Understand Python's built-in data structures (lists, sets, dictionaries, tuples).

Resources:

- 1. Python Basics: [Real Python](https://realpython.com/)
- 2. Big O Complexity: [Big-O Cheat Sheet](https://www.bigocheatsheet.com/)

Practice:

- Implement factorial and Fibonacci (iterative and recursive).
- Write a script to calculate time complexity manually for basic algorithms.

Week 2: Arrays and Strings

Goals:

- Master manipulation techniques for arrays and strings.
- Solve problems on searching, sorting, and sliding window.

Resources:

1. Arrays and Strings: [GeeksforGeeks - Arrays](https://www.geeksforgeeks.org/arrays/),

[Strings](https://www.geeksforgeeks.org/string-data-structure/)

2. LeetCode Practice: [LeetCode - Arrays Problems](https://leetcode.com/problemset/all/)

Practice:

- Problems like "Two Sum	', "Reverse a String",	"Longest Substring	Without Repeating Characters	"-
--------------------------	------------------------	--------------------	------------------------------	----

Week 3: Stacks and Queues

Goals:

- Understand stack/queue operations and use cases.
- Solve problems involving parentheses, next greater element, etc.

Resources:

- 1. Stack and Queue Basics: [Programiz Stack and Queue](https://www.programiz.com/)
- 2. Common Problems: [LeetCode Stacks/Queues](https://leetcode.com/problemset/all/)

Practice:

- Implement a stack and queue using arrays and linked lists.
- Solve "Valid Parentheses", "Min Stack".

Week 4: Linked Lists

Goals:

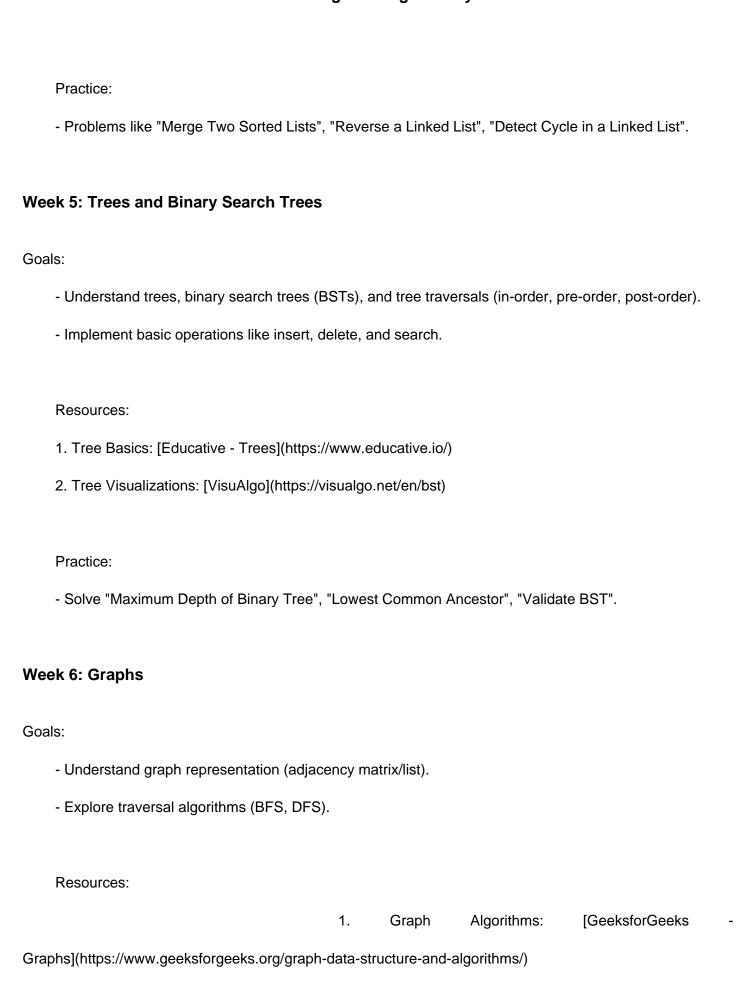
- Implement singly and doubly linked lists.
- Practice traversal, insertion, deletion, and reversing a list.

Resources:

1. Linked List Basics: [GeeksforGeeks - Linked

Lists](https://www.geeksforgeeks.org/data-structures/linked-list/)

2. Video Tutorial: [FreeCodeCamp Linked Lists](https://youtu.be/WwfhLC16bis)



2. Interactive Visualizations: [CS Academy Graphs](https://csacademy.com/app/graph_editor/)
Practice:
- Solve "Number of Islands", "Clone Graph", "Shortest Path in a Grid".
Week 7: Advanced Topics
Goals:
- Explore heaps (priority queues) and hash tables.
- Solve dynamic programming (DP) problems and backtracking.
Resources:
1. Heaps: [GeeksforGeeks - Heaps](https://www.geeksforgeeks.org/heap-data-structure/)
2. DP Basics: [Tutorial](https://www.geeksforgeeks.org/dynamic-programming/)
3. Backtracking: [FreeCodeCamp](https://www.freecodecamp.org/news/backtracking-explained/)
Practice:
- Problems like "Top K Frequent Elements", "House Robber", "N-Queens Problem".
Week 8: System Design & Mock Interviews
Goals:
- Learn about basic system design concepts and scalability.
- Practice mock coding interviews to solidify knowledge.
Resources:

		1.	System	Design:	[Grokking	the	System	Design	
Inte	Interview](https://www.educative.io/courses/grokking-the-system-design-interview)								
	2. Coding Platforms: [LeetCode](https://leetcode.com/), [HackerRank](https://www.hackerrank.com/)								
	Practice:								
	- Participate in mock interviews	s on [P	ramp](https:	//www.pram	p.com/).				

- Solve "Design TinyURL", "LRU Cache".