CoGrammar

Welcome to this session

Skills Bootcamp:

Techniques to Enhance Problem-Solving Thinking

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators
answering questions.

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Skills Bootcamp Full Stack Web Development

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. We will be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Skills Bootcamp Cloud Web Development

- For all **non-academic questions**, please submit a query: ***www.hyperiondev.com/support***

- **Report a safeguarding incident: *www.hyperiondev.com/safeguardreporting***

- We would love your feedback on lectures: ***Feedback on Lectures.***

- Find all the lecture **content** in your **Lecture Backpack** on GitHub.

- If you are hearing impaired, kindly use your computer's function through Google chrome to enable captions.

Skills Bootcamp Progression Overview

✅ Criterion 1 - Initial Requirements

Specific achievements within the first two weeks of the program.

To meet this criterion, students need to, by no later than **01 December 2024 (C11)** or **22 December 2024 (C12):**

- **Guided Learning Hours** (GLH): Attend a minimum of 7-8 GLH per week (lectures, workshops, or mentor calls) for a total minimum of **15 GLH**.
- **Task Completion:** Successfully complete the **first 4 of the assigned tasks**.

✅ Criterion 2 - Mid-Course Progress

Progress through the successful completion of tasks within the first half of the program.

To meet this criterion, students should, by no later than **12 January 2025 (C11)** or **02 February 2025 (C12):**

- **Guided Learning Hours** (GLH): Complete at least **60 GLH**.
- **Task Completion :** Successfully complete the **first 13 of the assigned tasks**.

CoGrammar

Skills Bootcamp Progression Overview

✅ Criterion 3 – End-Course Progress

Showcasing students' progress nearing the completion of the course.

To meet this criterion, students should:

- **Guided Learning Hours** (GLH)**:** Complete the **total minimum required GLH,** by the **support end date.**
- **Task Completion : Complete all mandatory tasks**, including any necessary resubmissions, by the end of the bootcamp, **09 March 2025 (C11)** or **30 March 2025 (C12)**.

✅ Criterion 4 - Employability

Demonstrating progress to find employment.

To meet this criterion, students should:

- **Record an Interview Invite:** Students are required to record proof of invitation to an interview by **30 March 2025 (C11)** or **04 May 2025 (C12)**.
  - **South Holland Students** are required to proof and interview by **17 March 2025.**
- **Record a Final Job Outcome :** Within 12 weeks post-graduation, students are required to record a job outcome.

CoGrammar

Stay Safe Series:

Mastering Online Safety One week at a Time

While the digital world can be a wonderful place to make education and learning accessible to all, it is unfortunately also a space where harmful threats like online radicalization, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

As a component of this BootCamp the *Stay Safe Series* will guide you through essential measures in order to protect yourself & your community from online dangers, whether they target your privacy, personal information or even attempt to manipulate your beliefs.

Trustworthy Websites:
How to Spot Secure Sites

- Look for the padlock.
- Check if there is a valid SSL/TLS certificate.
- Look for a site seal.
- Check if the URL is legitimate.
- Pop-up and Redirection ads are a red flag.

Why is problem solving important in Computer Science?

Which of the following is a common problem-solving framework?

A. UPER (Understand, Plan, Execute, Review)
B. Binary Search Tree
C. Object-Oriented Programming (OOP)
D. Machine Learning Models

Problem-Solving in Computer Science and Data Science

- What was the most challenging programming problem you've faced recently?

CoGrammar

Learning Outcomes

- Explain the importance of problem-solving in computer science and data science.
- Identify mental frameworks for breaking down complex problems.
- Develop logical reasoning and pattern recognition skills.
- Apply problem-solving techniques to real-world scenarios.

Lecture Overview

- → Introduction to Why Problem Solving Matters
- → Break (15 mins)
- → Pattern recognition and real-world applications
- → Interactive challenge and summary

Why Problem-Solving Matters

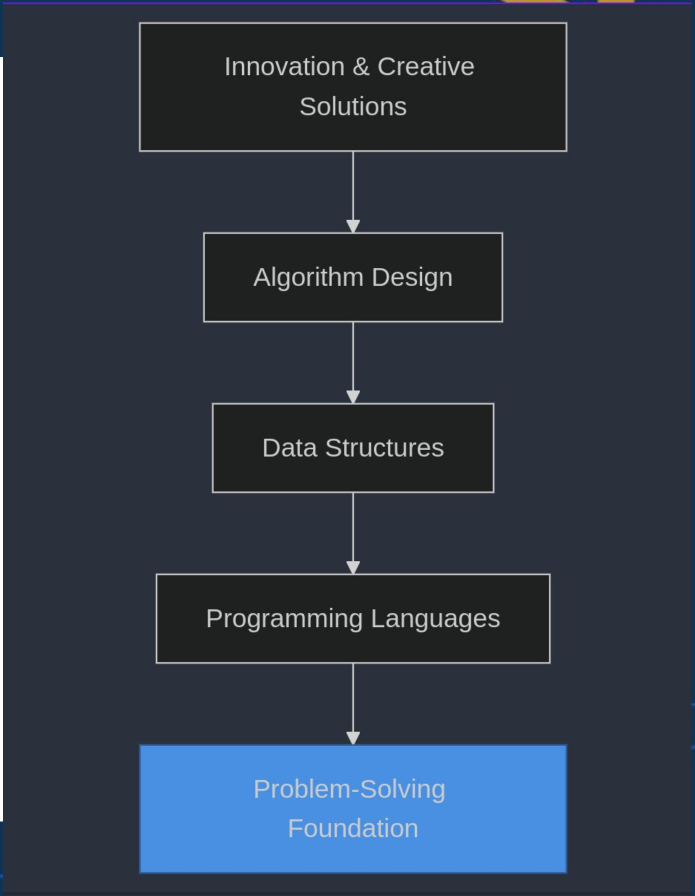
- Imagine you're building a house.
  - Data structures and algorithms are your tools and materials
  - Problem-solving is your architectural blueprint.
- Without problem-solving, even the best tools won't help you build effectively.

CoGrammar

Why Problem-Solving Matters

- Software Development: Debugging complex systems
- Data Science: Cleaning and transforming messy data
- System Design: Optimizing for scale and performance
- Machine Learning: Feature engineering and model selection

- The most elegant solution often comes from how you approach the problem, not just your coding skills.



Innovation & Creative Solutions

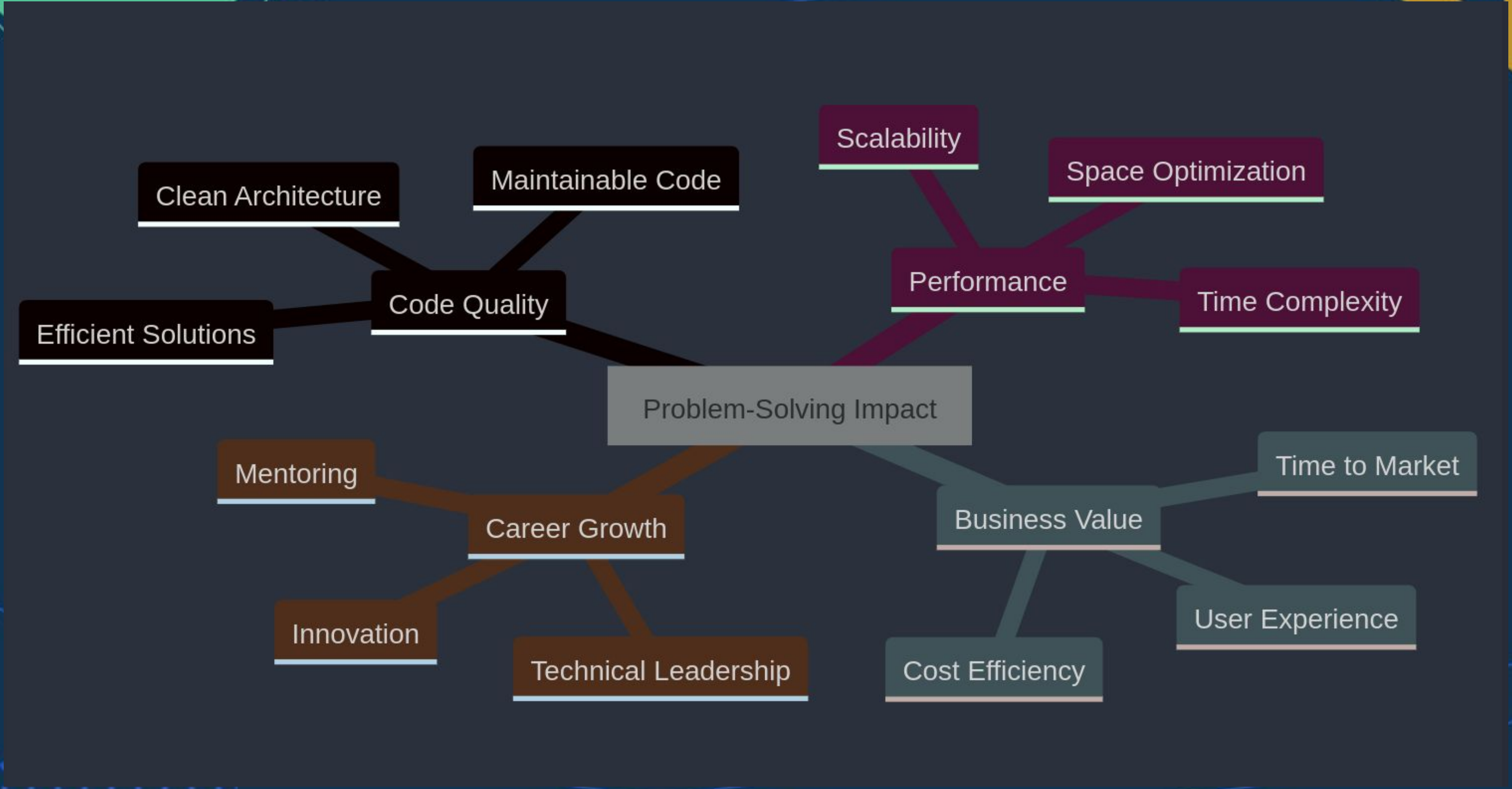
Algorithm Design

Data Structures

Programming Languages

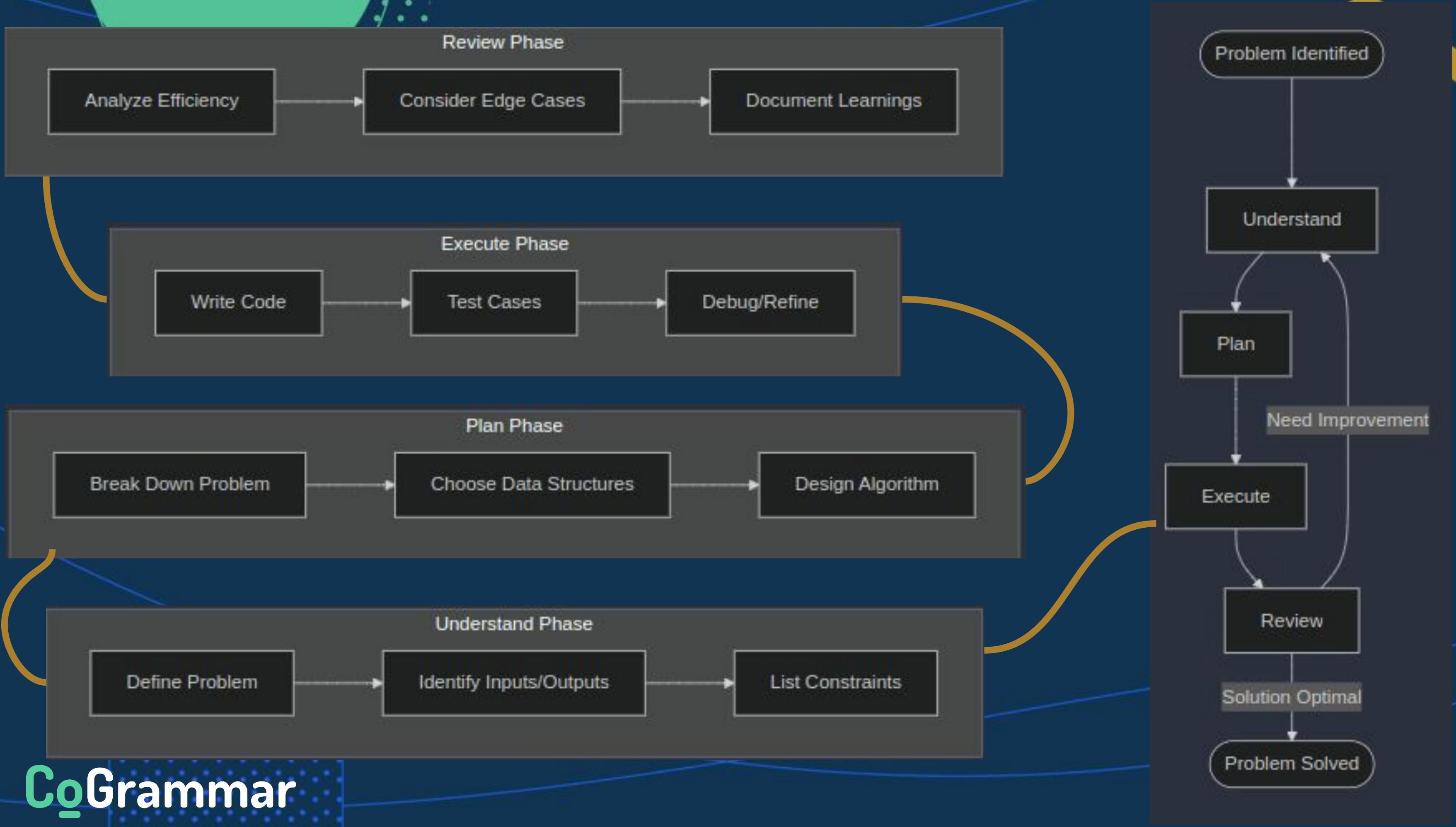
Problem-Solving Foundation

Problem-Solving Impact



CoGrammar

Mental Frameworks



CoGrammar

UPER Model

- Key Points:
  - Understand: Clarify the problem, inputs, outputs, and constraints.
  - Plan: Break into subproblems and choose suitable data structures.
  - Execute: Write modular, testable code.
  - Review: Analyze efficiency and optimi

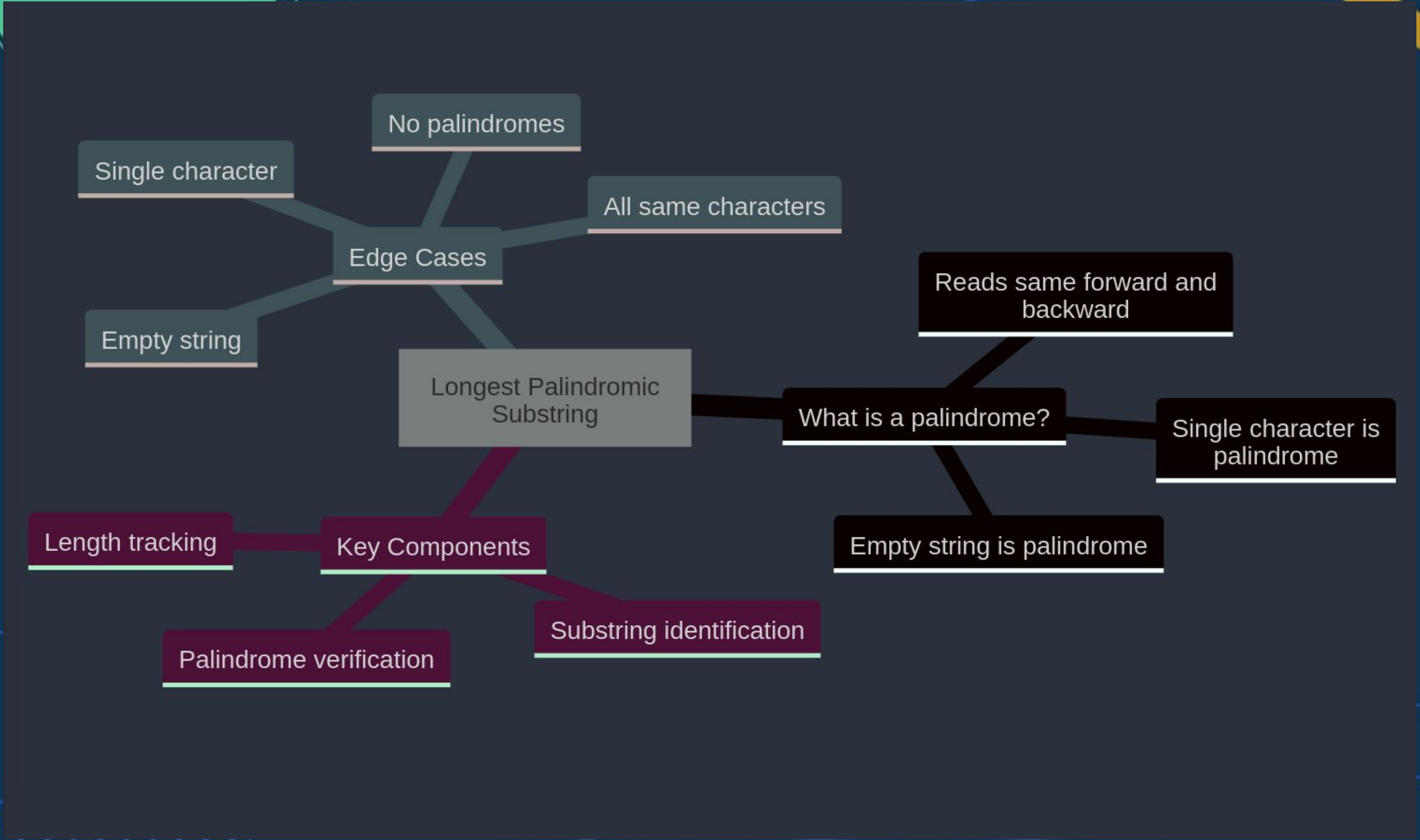
Let's take a break

CoGrammar

Problem Decomposition - Divide and Conquer in Action

- ❖ Find the longest palindromic substring

Mindmap



Longest Palindromic Substring

Edge Cases

- Single character
- No palindromes
- All same characters
- Empty string

What is a palindrome?

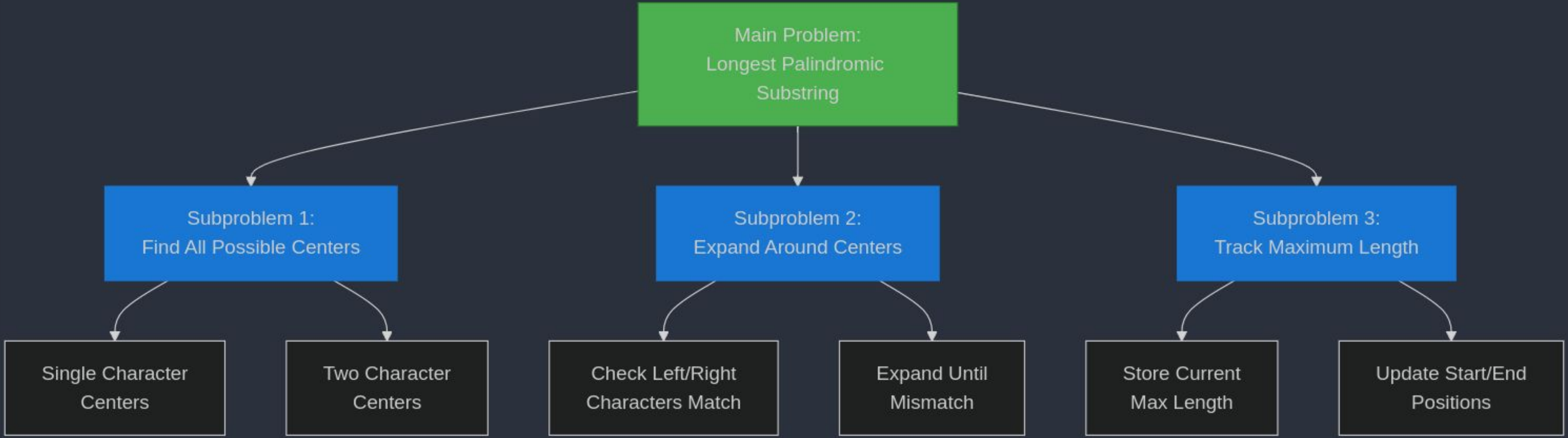
- Reads same forward and backward
- Single character is palindrome
- Empty string is palindrome

Key Components

- Length tracking
- Palindrome verification
- Substring identification

CoGrammar

Breaking Down the Problem



Main Problem:
Longest Palindromic Substring

Subproblem 1:
Find All Possible Centers

- Single Character Centers
- Two Character Centers

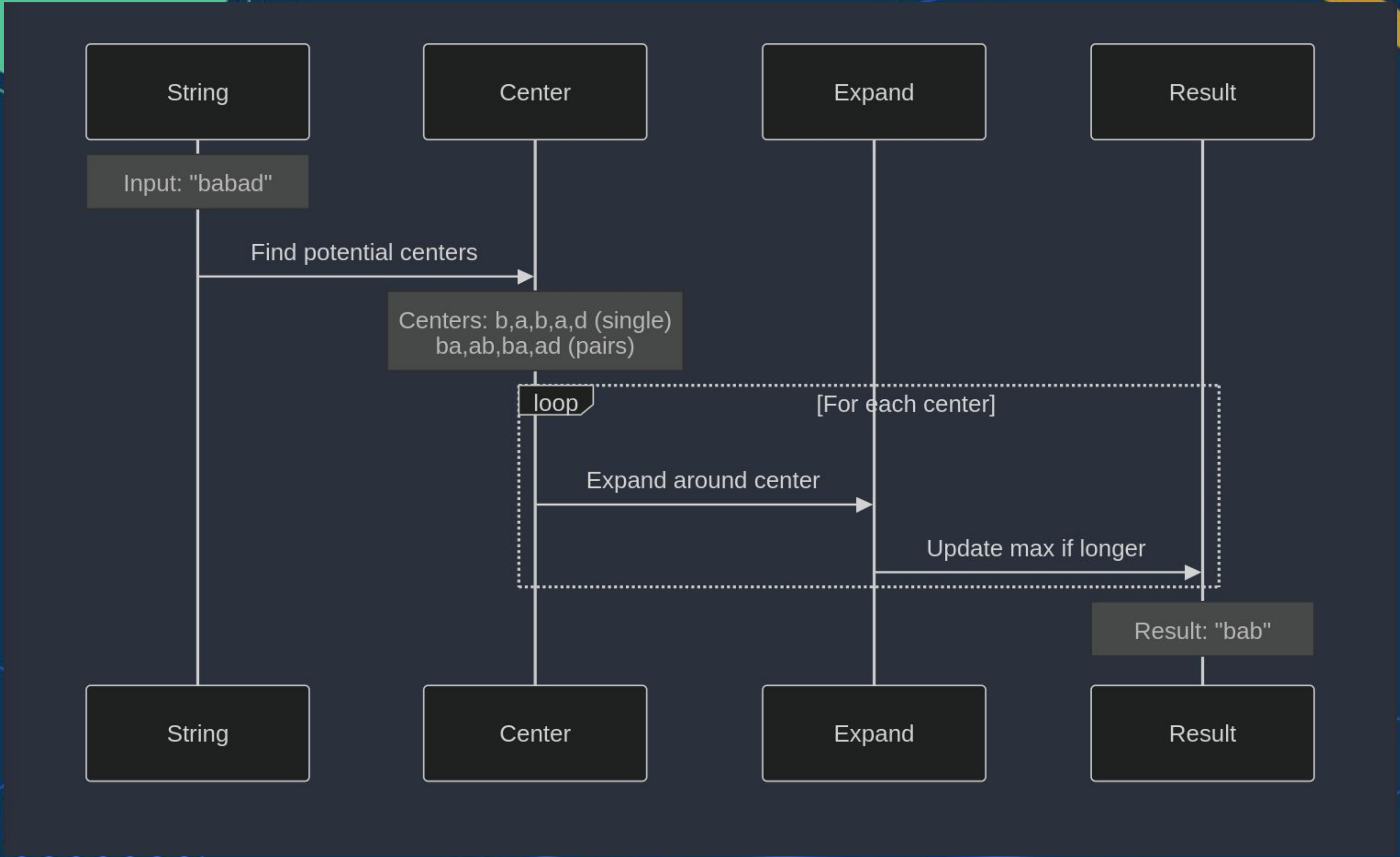
Subproblem 2:
Expand Around Centers

- Check Left/Right Characters Match
- Expand Until Mismatch

Subproblem 3:
Track Maximum Length

- Store Current Max Length
- Update Start/End Positions

Solution Approach



String

Center

Expand

Result

Input: "babad"

Find potential centers

Centers: b,a,b,a,d (single)
ba,ab,ba,ad (pairs)

loop

[For each center]

Expand around center

Update max if longer

Result: "bab"

String

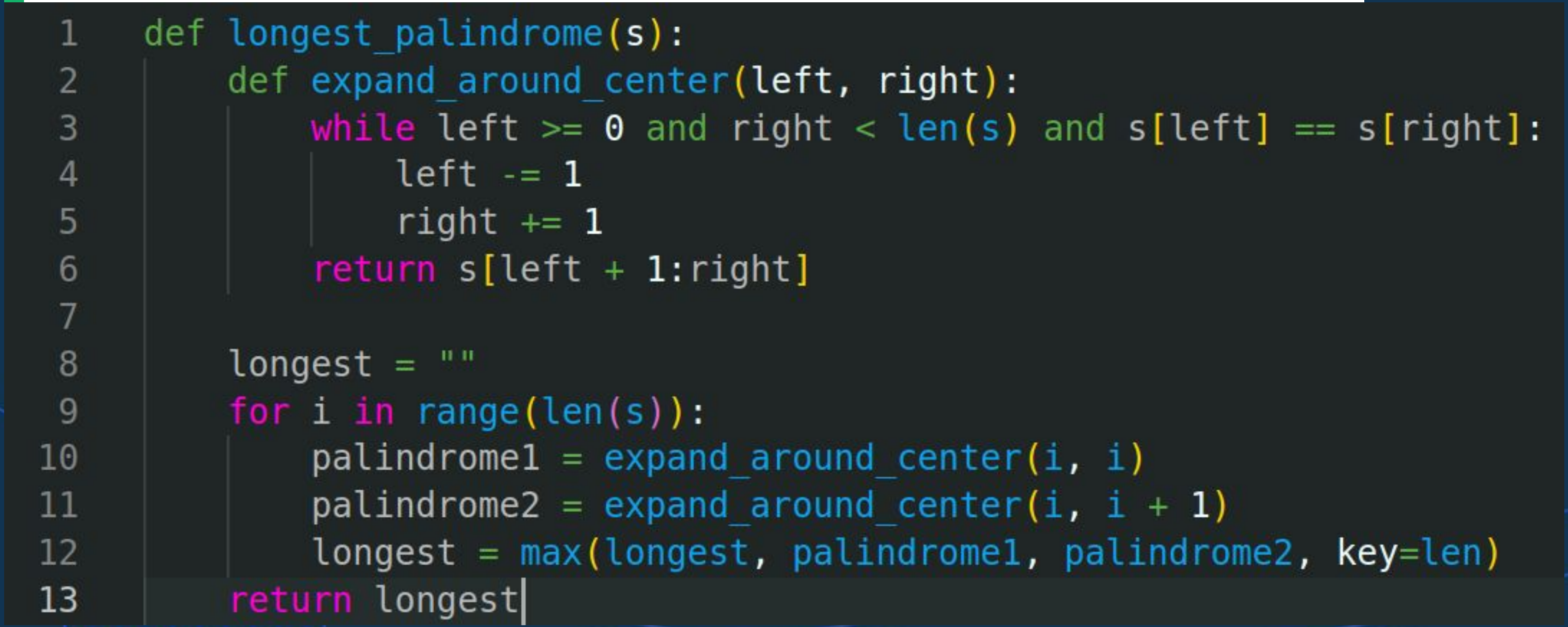
Center

Expand

Result

Problem Decomposition - Divide and Conquer in Action

❖ Find the longest palindromic substring



```python
def longest_palindrome(s):
    def expand_around_center(left, right):
        while left >= 0 and right < len(s) and s[left] == s[right]:
            left -= 1
            right += 1
        return s[left + 1:right]

    longest = ""
    for i in range(len(s)):
        palindrome1 = expand_around_center(i, i)
        palindrome2 = expand_around_center(i, i + 1)
        longest = max(longest, palindrome1, palindrome2, key=len)
    return longest
```

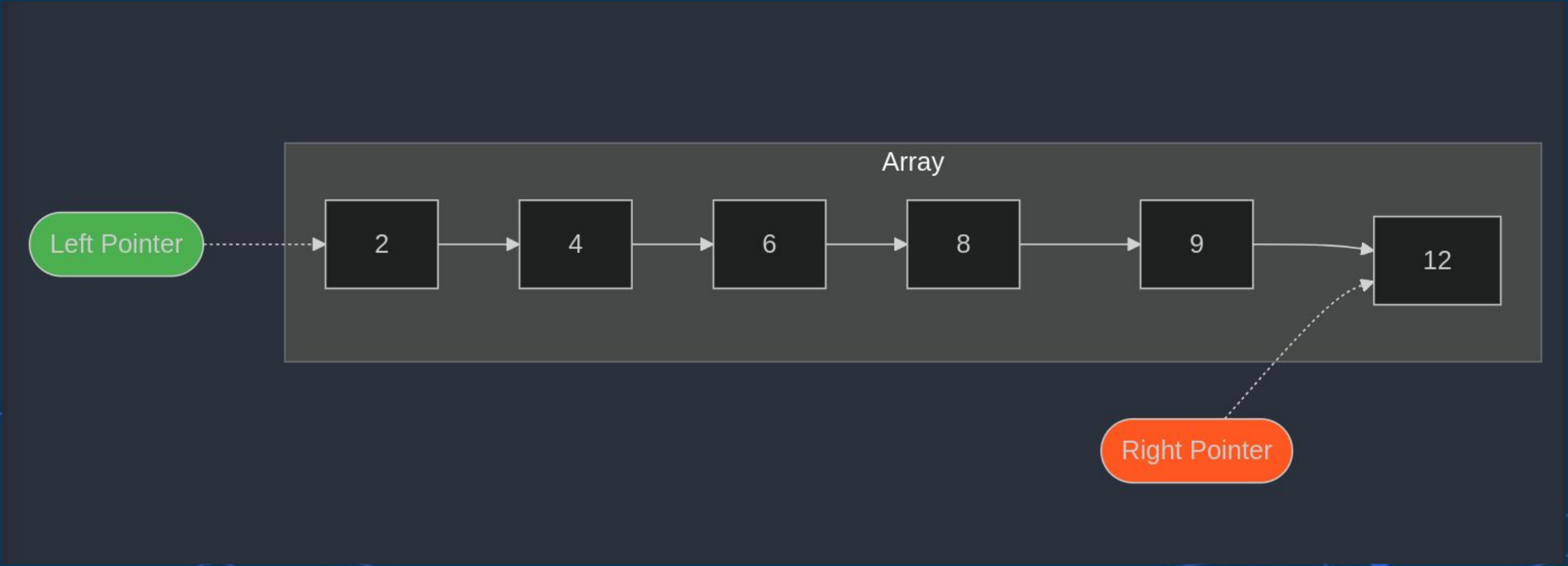
Pattern Recognition in Problem Solving

CoGrammar

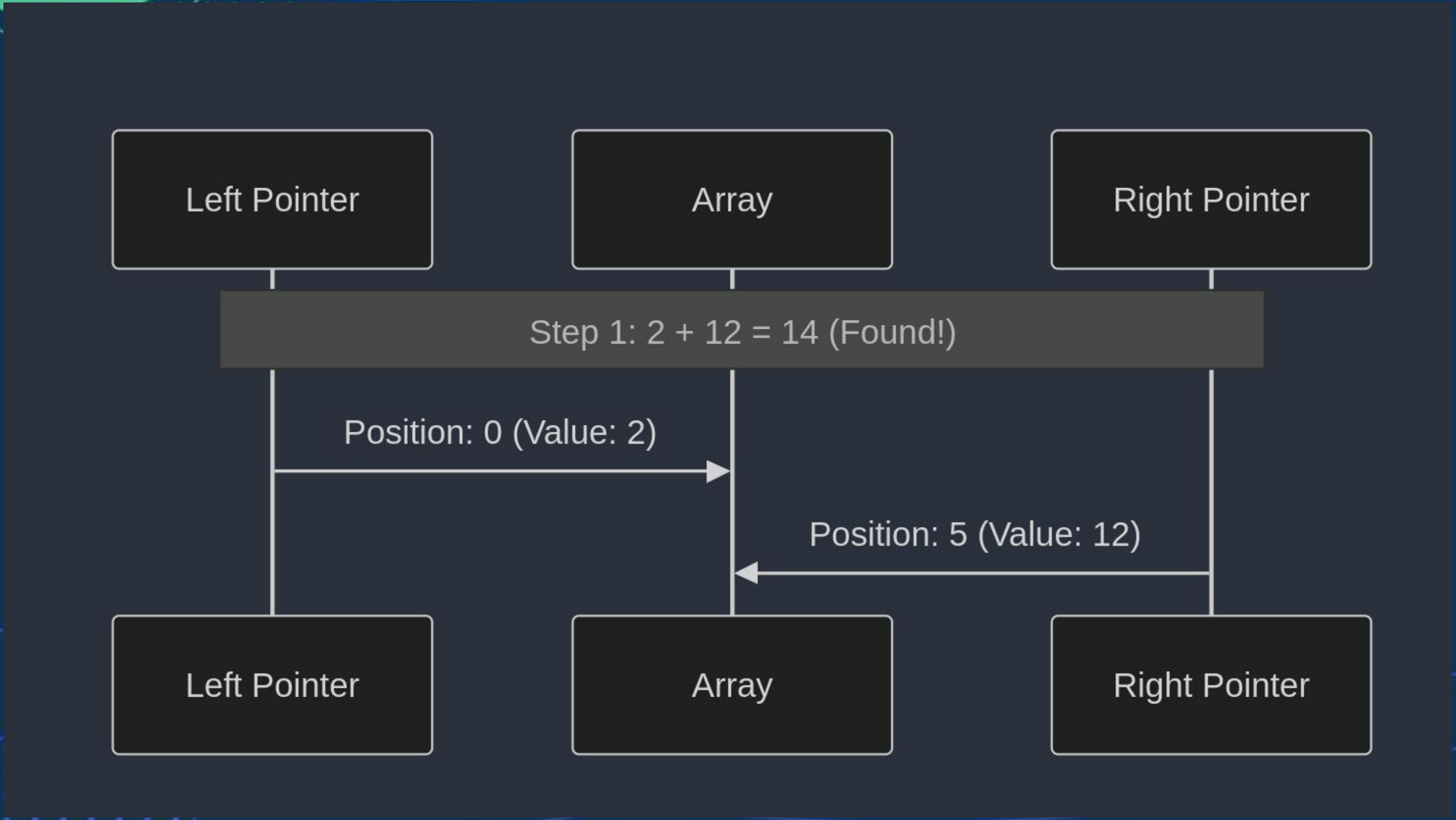
Identify Common Patterns

- Two-pointer technique
- Sliding window
- Binary search
- Dynamic programming

Two-Pointer Technique: Finding Pair Sum in Sorted Array

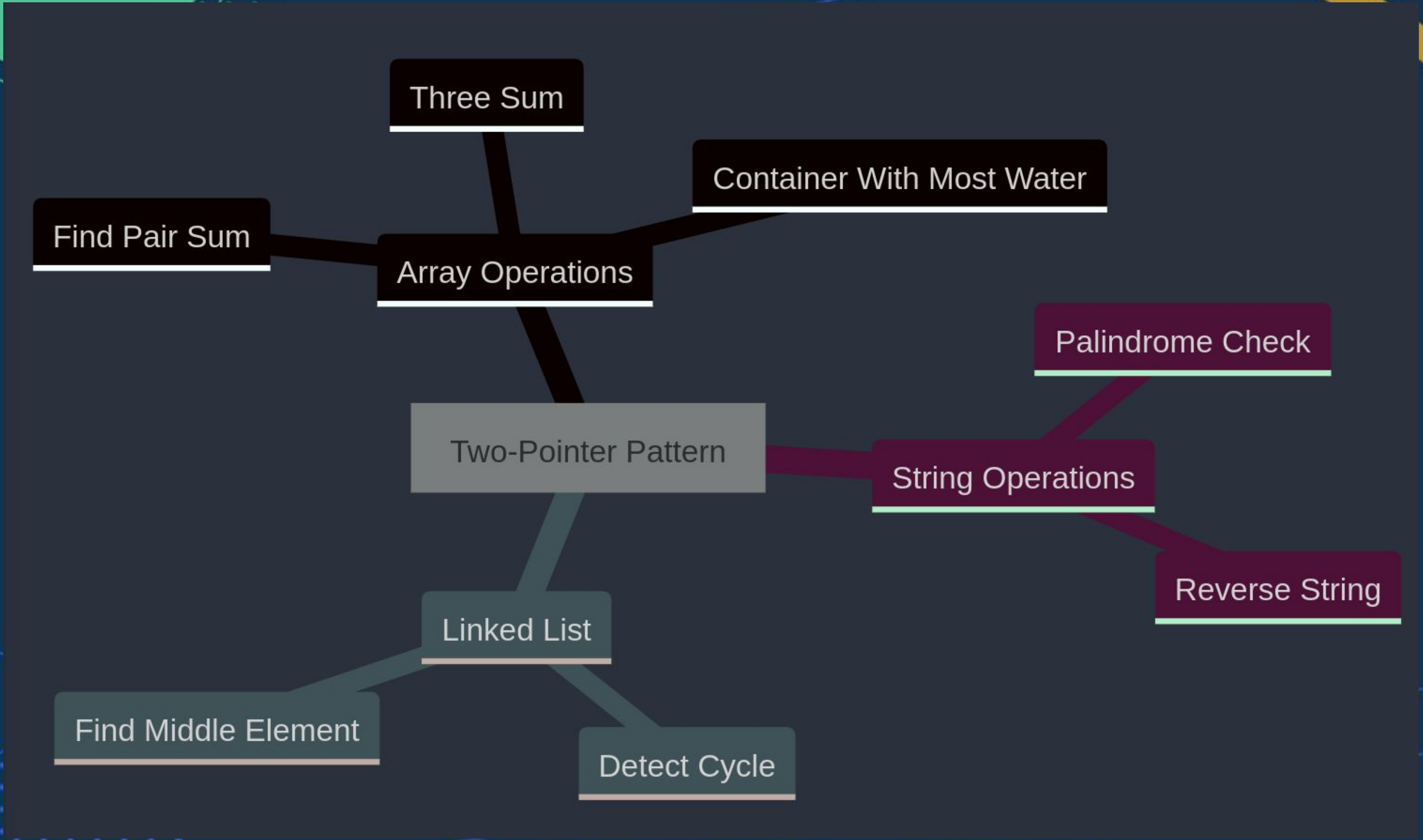


Target Sum = 14



CoGrammar

Common Applications



Three Sum

Container With Most Water

Find Pair Sum

Array Operations

Palindrome Check

Two-Pointer Pattern

String Operations

Reverse String

Linked List

Find Middle Element

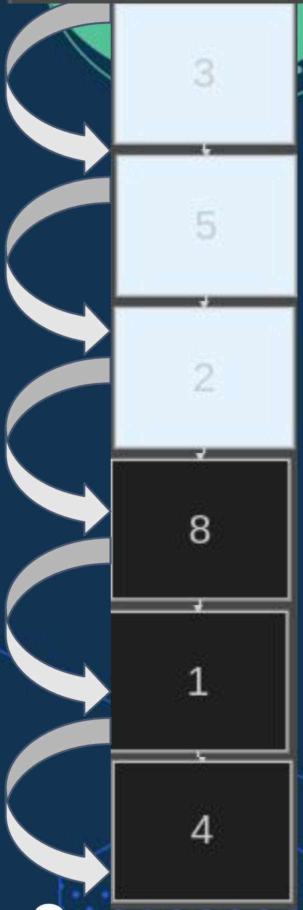
Detect Cycle

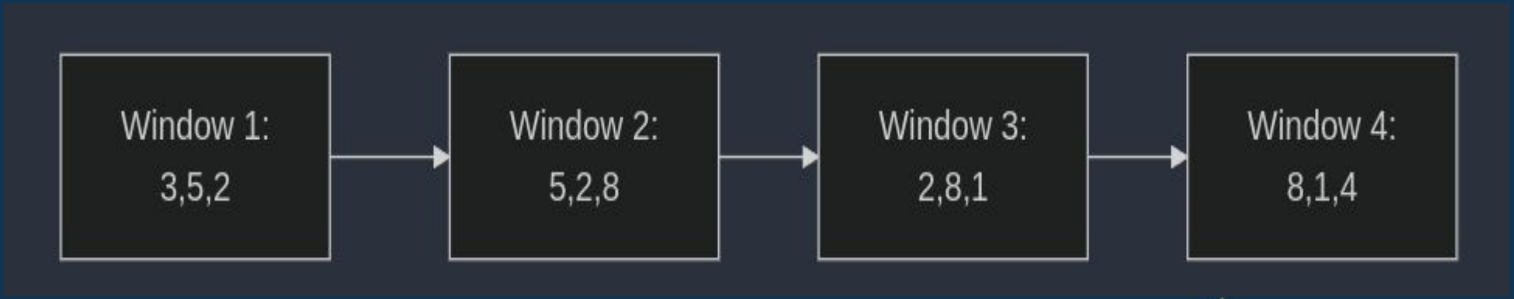
Sliding Window

- The sliding window pattern maintains a window of elements that slides through the array or string, updating results based on window contents.

Sliding Window

Window Slides Right





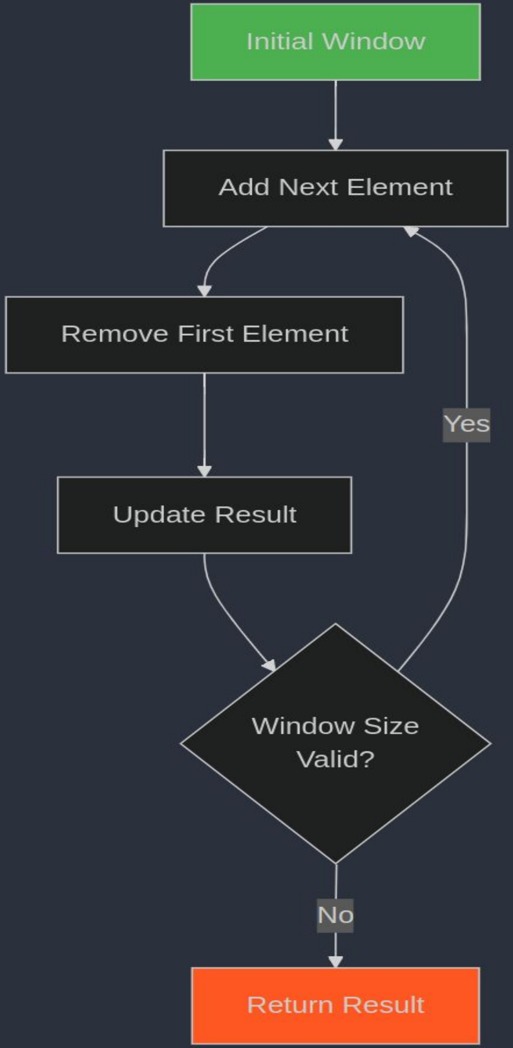
Window 1:
3,5,2

Window 2:
5,2,8

Window 3:
2,8,1

Window 4:
8,1,4

Window Operations

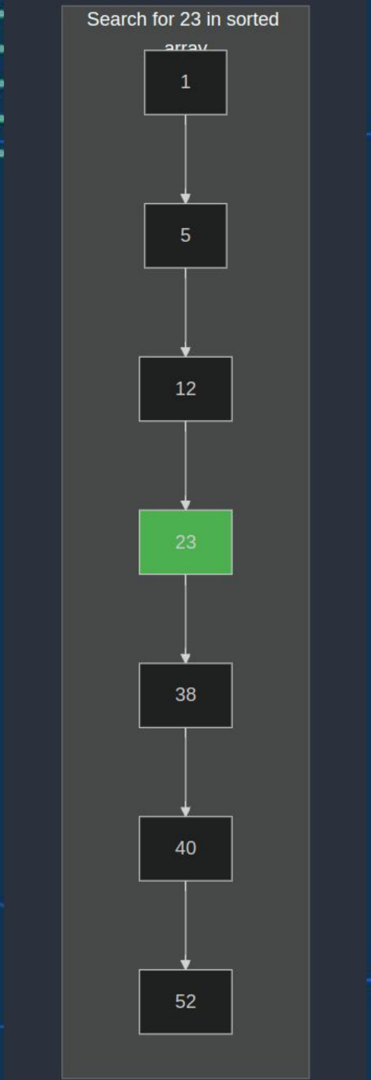


CoGrammar

Binary Search

- Binary search divides the search space in half at each step, eliminating half of the remaining elements based on a comparison.

Binary Search



Search for 23 in sorted array

1

5

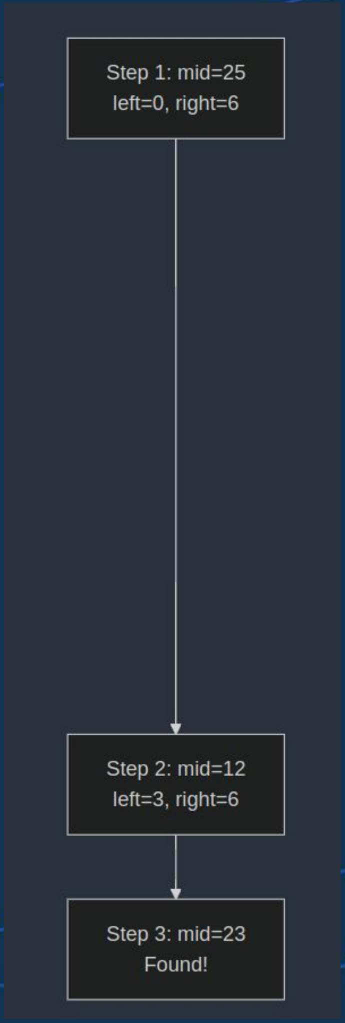
12

23

38

40

52

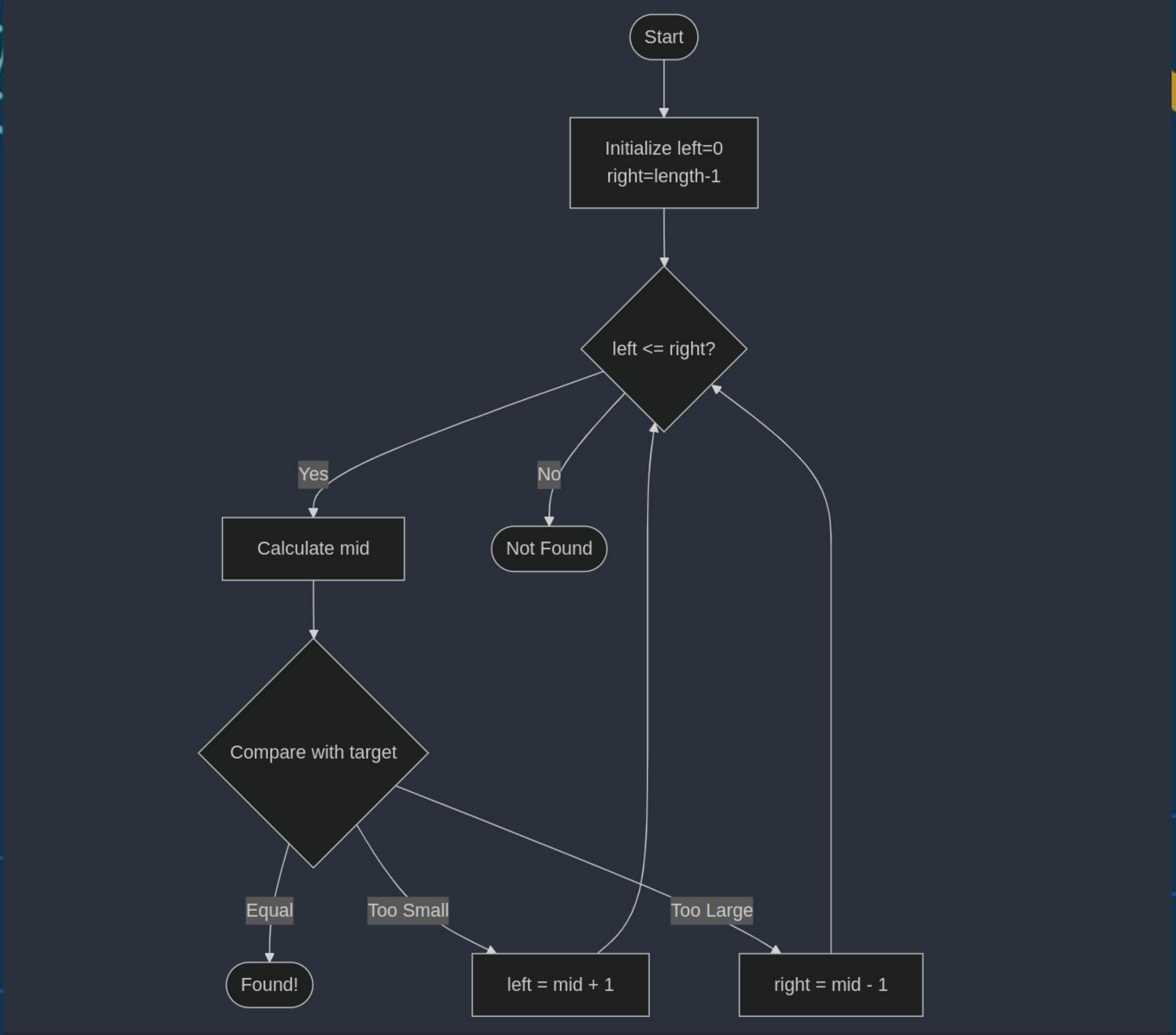


Step 1: mid=25
left=0, right=6

Step 2: mid=12
left=3, right=6

Step 3: mid=23
Found!

Binary Search

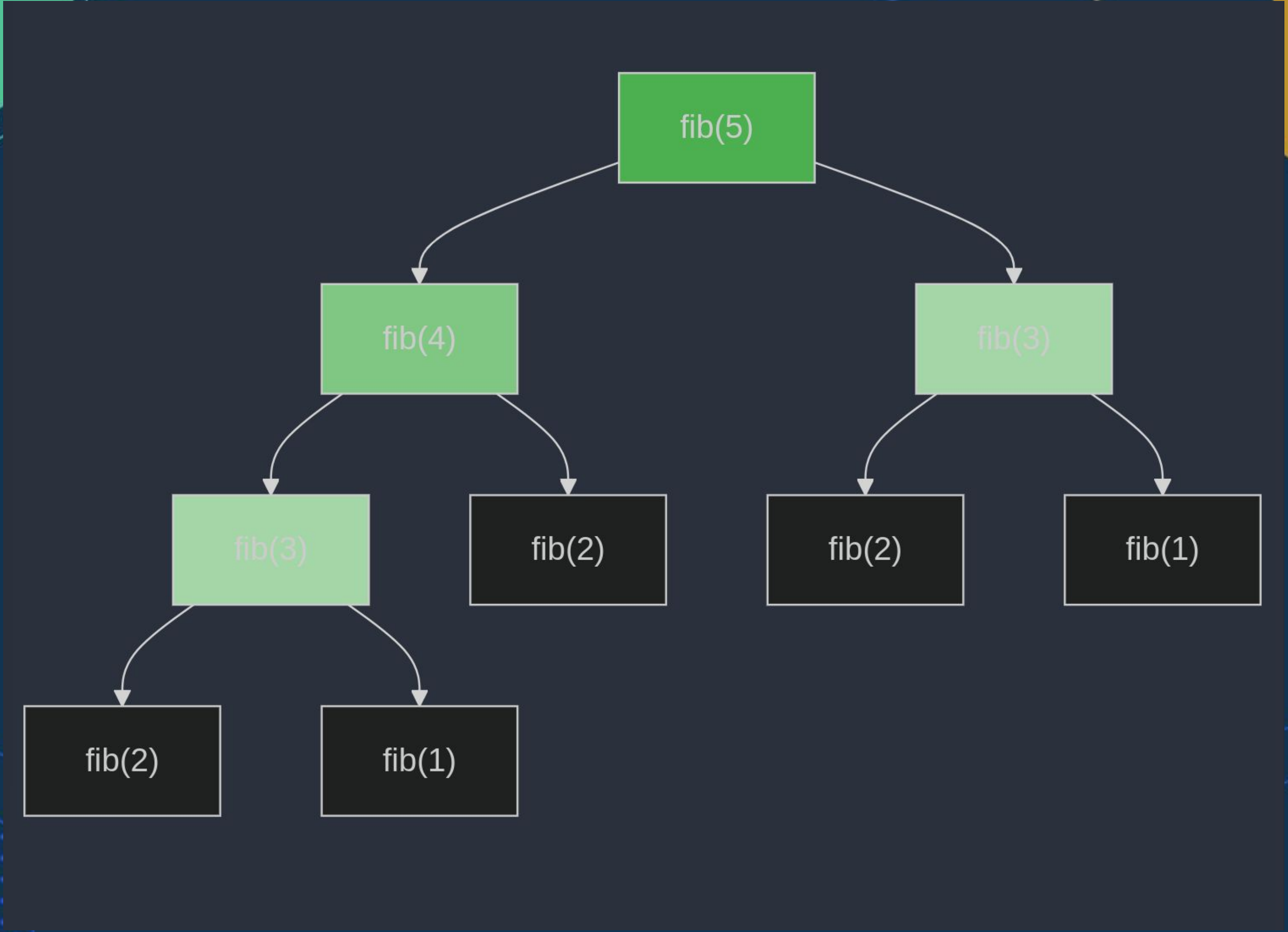


CoGrammar

Dynamic Programming

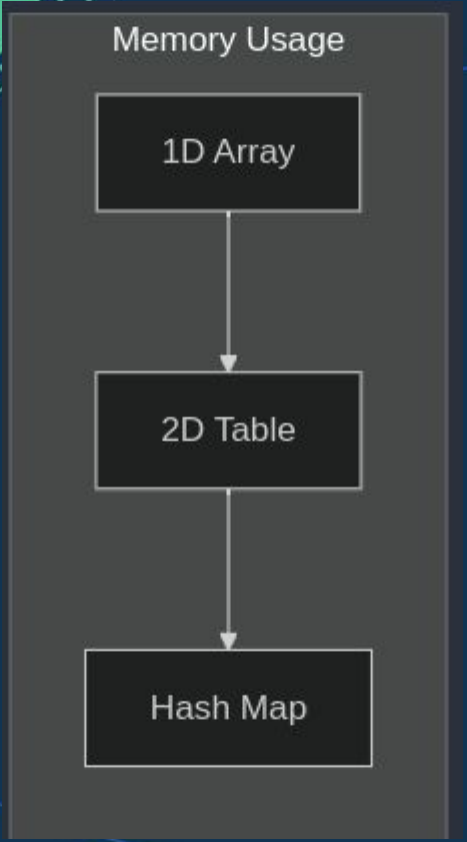
- Dynamic programming breaks down complex problems into simpler subproblems and stores their results to avoid redundant calculations.

DP

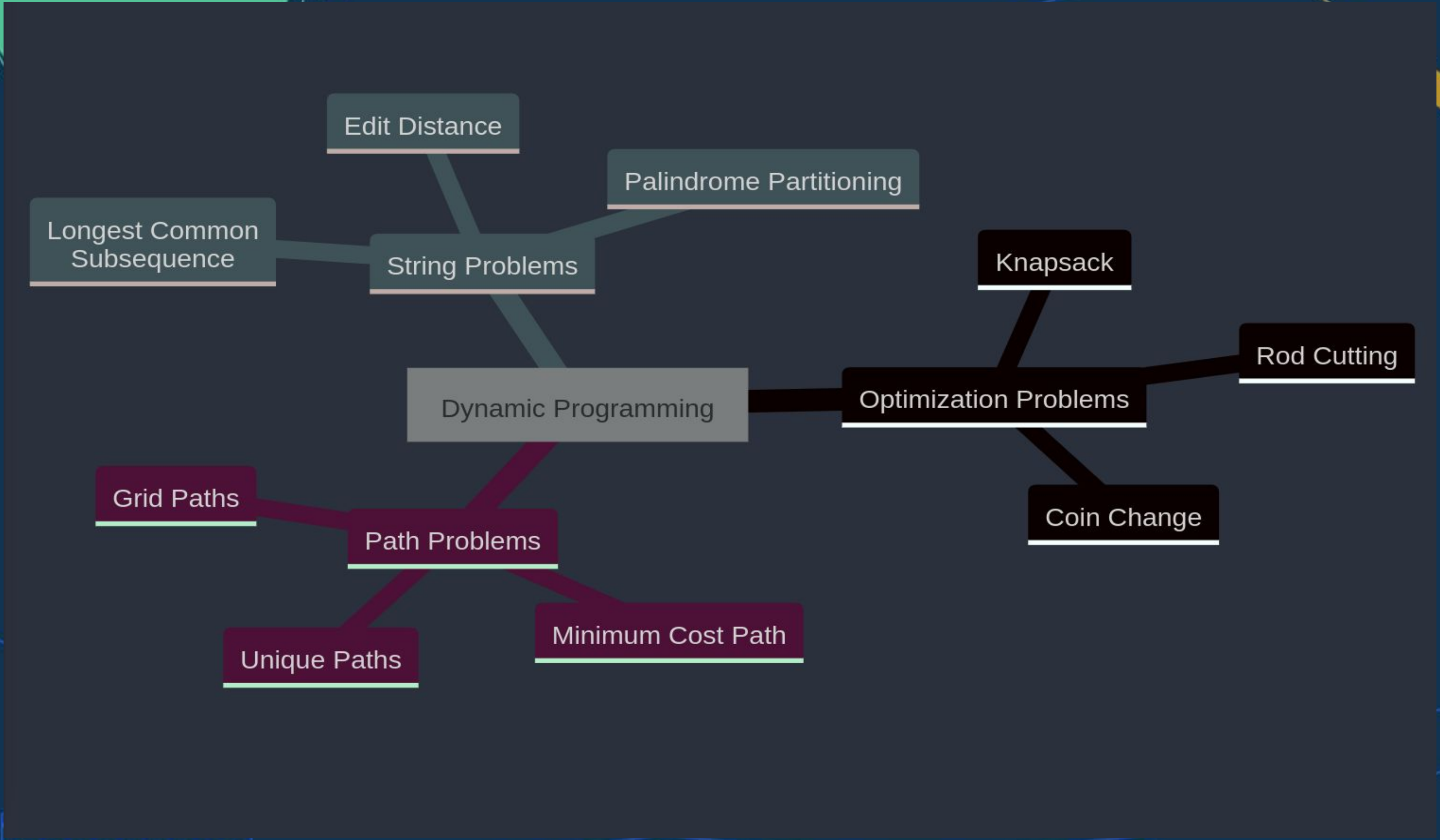


CoGrammar

DP Solving steps



DP Solution Steps
Unsupported markdown: list
Unsupported markdown: list
Unsupported markdown: list
Unsupported markdown: list
Unsupported markdown: list


Edit Distance
Palindrome Partitioning
Longest Common Subsequence
String Problems
Knapsack
Rod Cutting
Dynamic Programming
Optimization Problems
Coin Change
Grid Paths
Path Problems
Minimum Cost Path
Unique Paths

Real-World Application

- **Case Study: Optimising a Data Pipeline**
- Scenario:
  - You have a slow data pipeline processing millions of records
  - Apply UPER to break down the pipeline and propose optimizations.

Applying Everything Learned

- Design a caching system for a web app that handles millions of requests efficiently
- Steps:
  - Apply UPER.
  - Identify patterns (LRU, LFU caching).
  - Consider memory and speed trade-offs.

CoGrammar

Review and Next Steps

- Key Takeaways:
  - Problem-solving frameworks provide structure.
  - Practice pattern recognition and decomposition.
  - Real-world applications show the value of systematic approaches.
- Continuing Your Learning Journey
  - LeetCode
  - Project Euler
  - Open-source contributions

Additional Resources

- Python Documentation:
  - https://python.org
- Algorithm Visualizer:
  - https://visualgo.net
- Practice Problems:
  - https://leetcode.com

CoGrammar

Which framework involves breaking a problem into smaller tasks?

A. Divide and Conquer
B. Greedy Algorithm
C. Brute Force
D. Dynamic Programming

Problem-solving is only useful for competitive programming.

A. True

B. False

Questions and Answers

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education