CoGrammar

Welcome to this session

Skills Bootcamp:

Introduction to Algorithms and Problem-Solving Methodologies

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.

Don’t Take the Bait: How to Spot Phishing Scams

- Check the Sender's Email Address
- Look for Generic Greetings
- Be Wary of Urgent Language
- Hover Over Links
- Inspect Attachments Carefully
- Look for Spelling and Grammar Errors
- Verify with the Source
- Use Multi-Factor Authentication
- Stay Informed
- Report Suspicious Emails

What is an Algorithm?

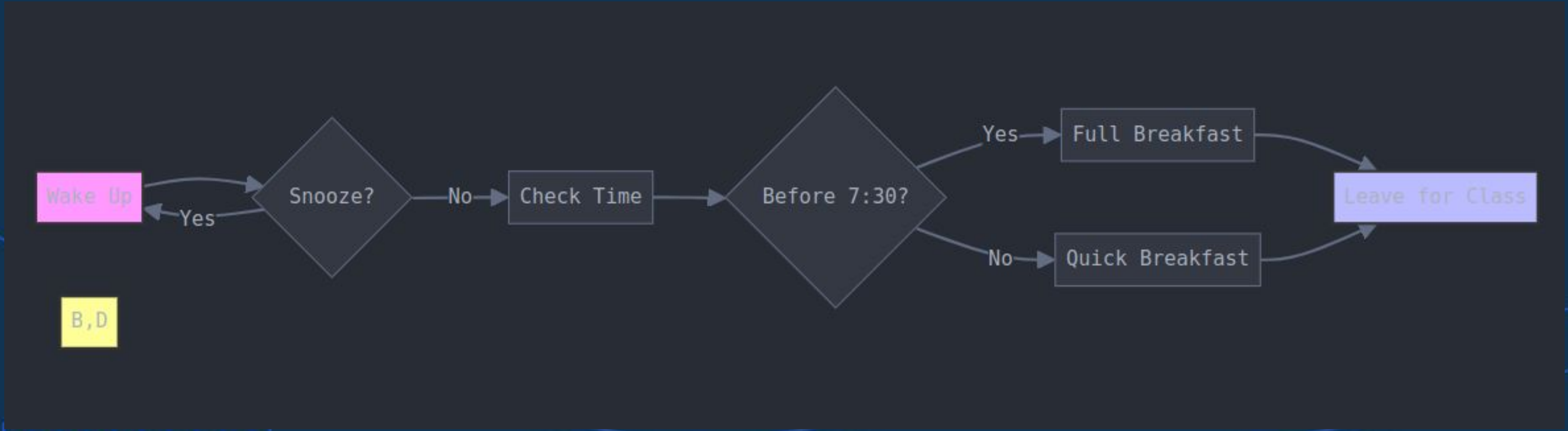
A. A detailed step-by-step procedure for solving a problem
B. A programming language
C. A hardware component
D. A design pattern

CoGrammar

Which of the following is a characteristic of a good algorithm?

A. Complexity and Length
B. Correctness, Efficiency, and Readability
C. Use of Multiple Programming Languages
D. Randomness in Output

- Think about your morning routine - from the moment you wake up until you arrive at class. How many decisions do you make, and what's your process for making them efficiently? For example, do you optimize your route to avoid traffic, or plan your breakfast to save time? This is actually an algorithm you've created! Could someone share their morning 'algorithm' and explain why they chose that specific sequence of steps?



CoGrammar

- What happens when something unexpected disrupts your routine?
- Have you ever tried to optimize or improve your morning sequence?
- How do you handle multiple constraints like time, resources, and preferences?

CoGrammar

Learning Outcomes

- Define and explain the concept of algorithms in computer science.
- Describe the characteristics of a good algorithm (correctness, efficiency, readability).
- Identify and utilise basic problem-solving methodologies such as divide-and-conquer, brute force, and greedy algorithms.
- Illustrate the process of translating a problem into a step-by-step algorithm.

Lecture Overview

- → Introduction to Algorithms
  - ◆ Fundamentals and characteristics
- → Break (15 mins)
- → Problem-Solving Methods
  - ◆ Different approaches
- → Assessment and Q&A

What is an Algorithm?

- An **algorithm** is:
  - A precise, step-by-step procedure for solving a problem
  - Comparable to a recipe for computers
- Key Characteristics:
  - Precise: Clear instructions
  - Unambiguous: No room for interpretation
  - Finite: Must terminate
- Examples in Daily Life:
  - Following a cooking recipe
  - Giving directions to a destination

Why are algorithms important?

- Foundation of Computer Programming:
  - Core to software development
- Efficiency in Problem-Solving:
  - Streamlines complex tasks
- Applications in Various Fields:
  - Data Processing
  - Search Engines
  - Financial Systems
  - Machine Learning

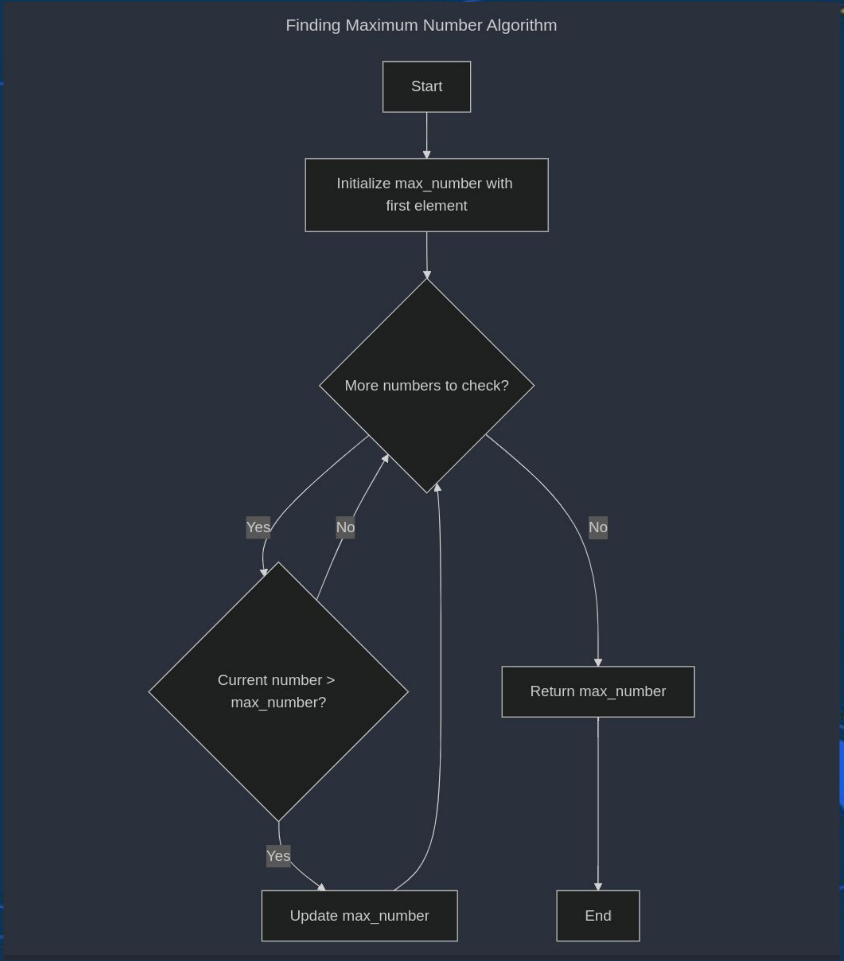
CoGrammar

Characteristics of Good Algorithms

The Three Pillars:

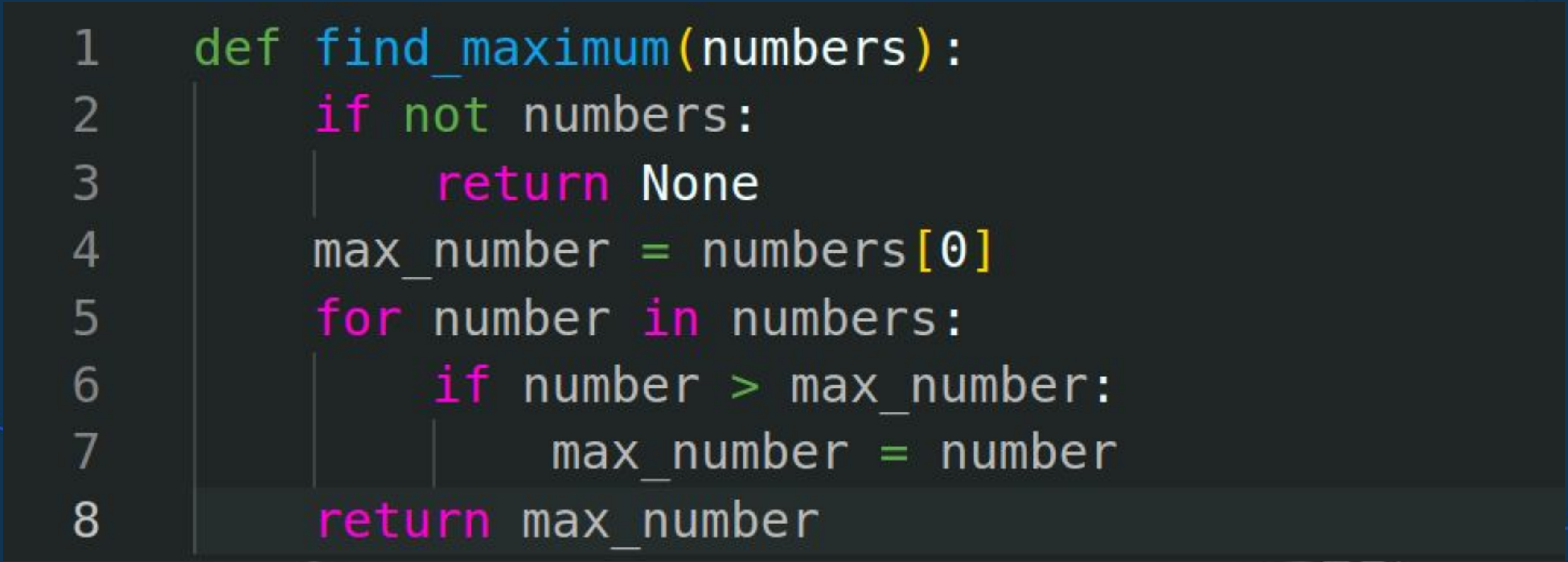
- ❖ Correctness:
  - ➢ Produces the correct output for all valid inputs
- ❖ Efficiency:
  - ➢ Optimizes resource use (time and memory)
- ❖ Readability:
  - ➢ Clear, understandable, and maintainable code

Example: Finding the Maximum Number


Finding Maximum Number Algorithm
Start
Initialize max_number with first element
More numbers to check?
Yes
No
No
Current number > max_number?
Yes
Return max_number
Update max_number
End

CoGrammar

Example: Finding the Maximum Number



```python
def find_maximum(numbers):
    if not numbers:
        return None
    max_number = numbers[0]
    for number in numbers:
        if number > max_number:
            max_number = number
    return max_number
```

CoGrammar

Algorithm Analysis

- Using find_maximum as an example:
  - Correctness:
    - Works for all valid cases
  - Efficiency:
    - Checks each number once
  - Readability:
    - Clear variable names and logic

CoGrammar

Common Problem Solving Methodologies

- ❖ Divide-and-Conquer
- ❖ Brute Force
- ❖ Greedy Algorithms

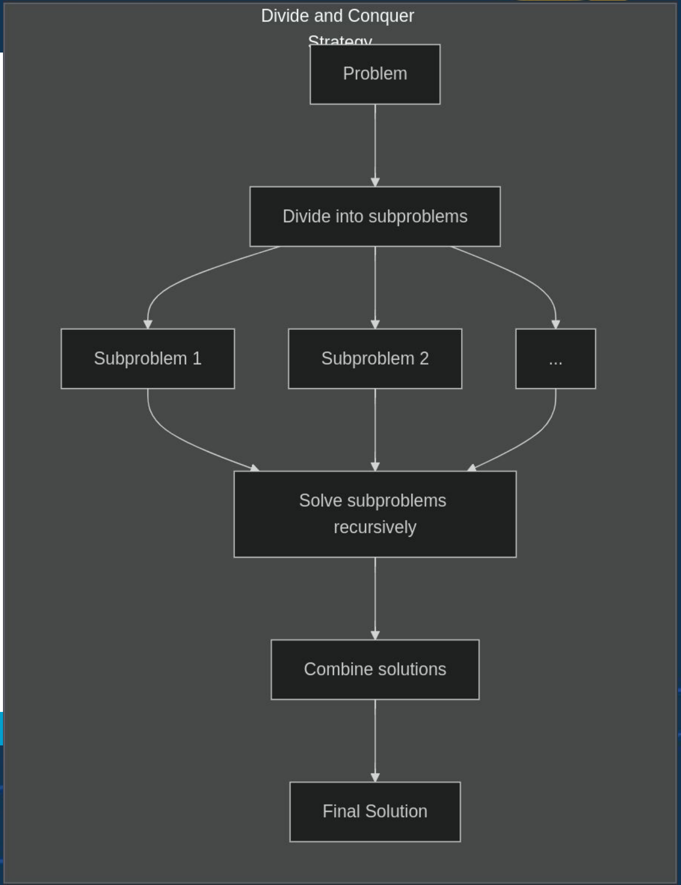
CoGrammar

Let's take a break

CoGrammar

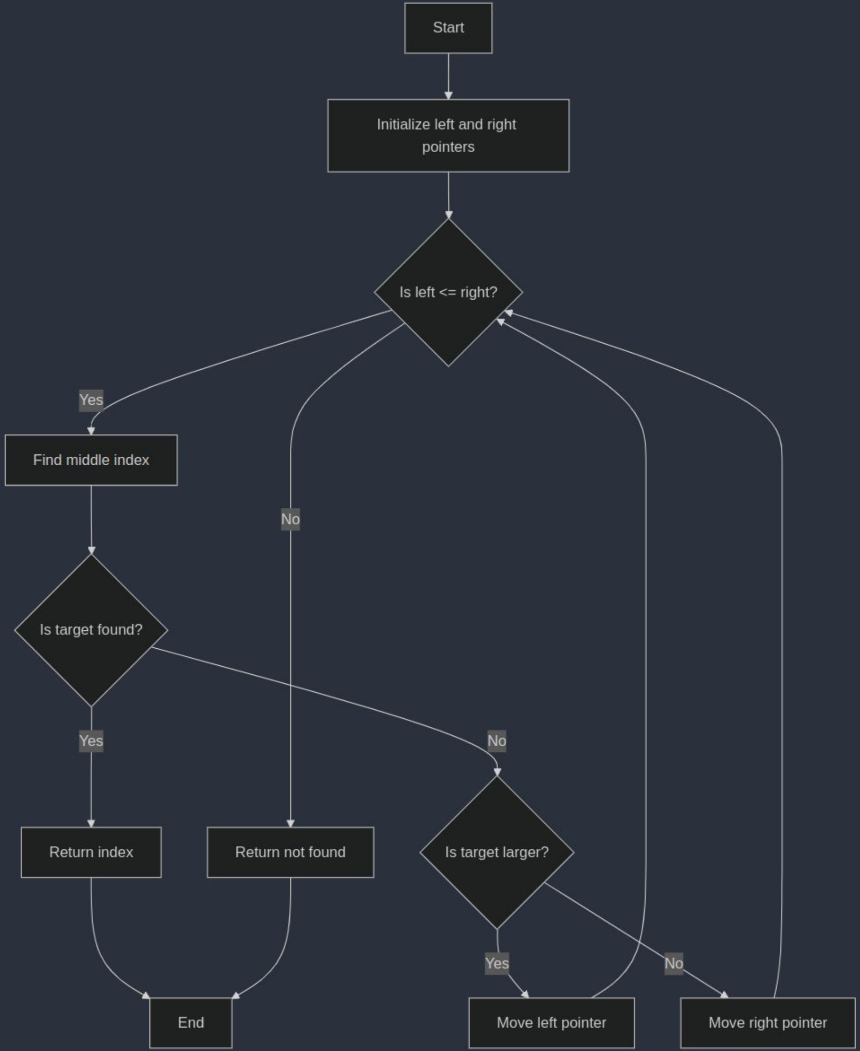
Divide and Conquer

- ❖ Break the problem into smaller subproblems
- ❖ Solve subproblems recursively
- ❖ Combine results for the final solution



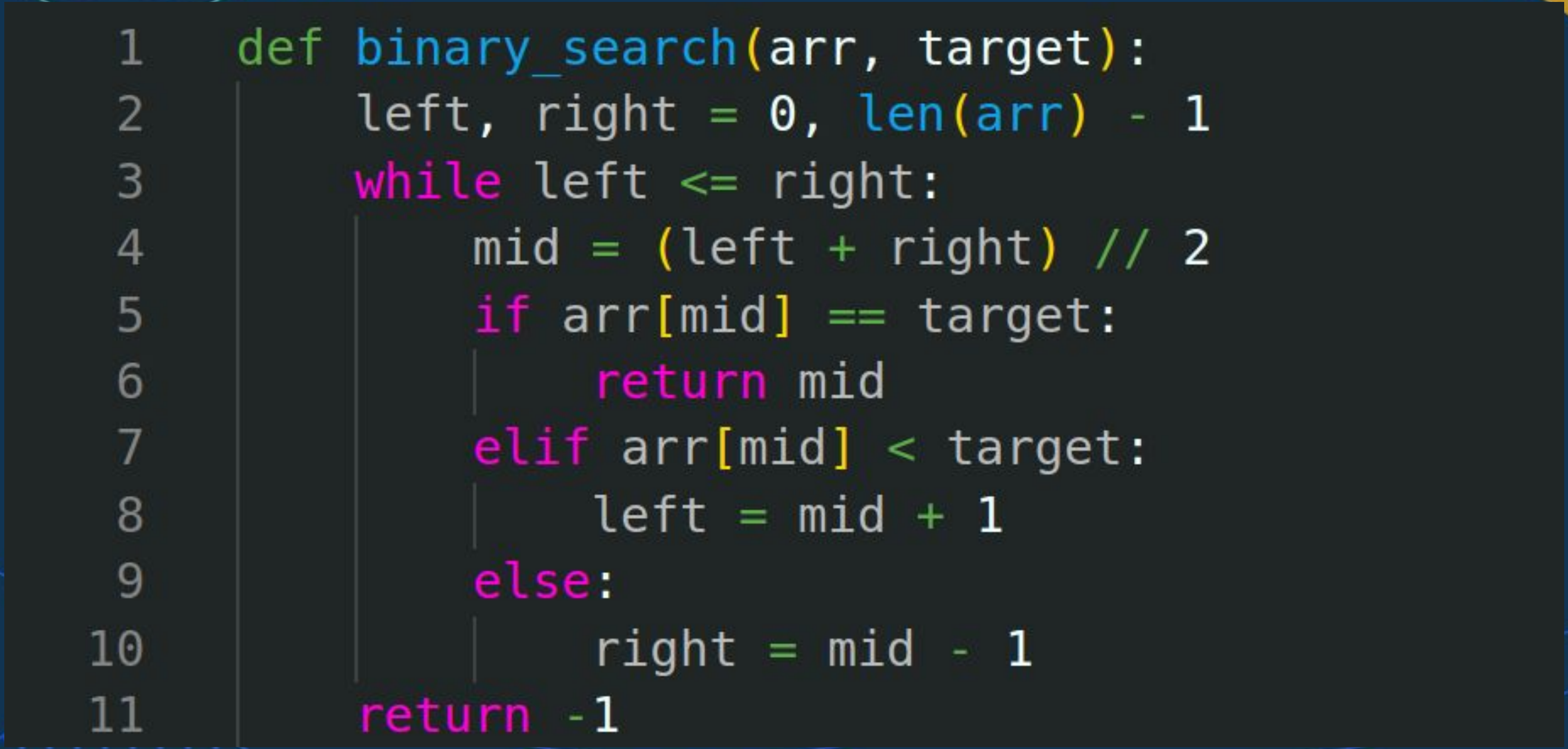
CoGrammar

Binary Search Example



CoGrammar

Binary Search Example



```python
def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

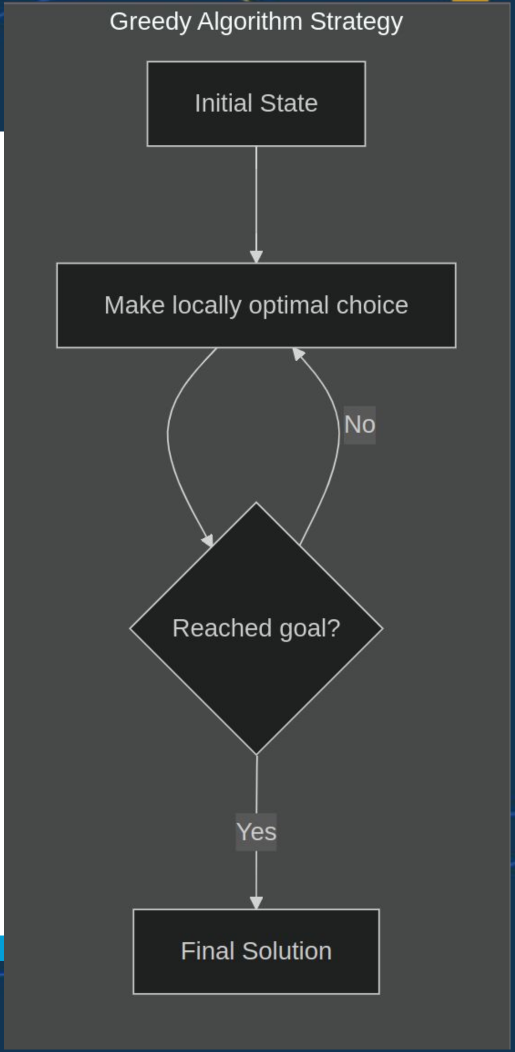
CoGrammar

Brute Force Method

- ❖ Definition:
  - ➢ Try all possible solutions
- ❖ Advantages:
  - ➢ Simple, guaranteed solution
- ❖ Disadvantages:
  - ➢ Inefficient for large data
- ❖ Best Used For:
  - ➢ Small datasets
  - ➢ Validating other algorithms
  - ➢ When no optimized solution exists

Greedy Algorithms

- ❖ Make the locally optimal choice at each step
- ❖ Assume it will lead to the globally optimal solution
- ❖ Examples:
  - ➢ Coin Change Problem
  - ➢ Huffman Coding
  - ➢ Activity Selection



CoGrammar

Real-World Application: Route Planning

- ❖ **Problem**: Find the shortest path between two points
- ❖ **Solutions**:
  - ➢ **Brute Force**: Try all routes
  - ➢ **Greedy**: Always move toward the destination
  - ➢ **Divide-and-Conquer**: Break down the journey into smaller steps

CoGrammar

Common Pitfalls in Algorithm Design

- Ignoring edge cases
- Premature optimization
- Overcomplicating the solution
- Failing to validate inputs
- Poor documentation

CoGrammar

Best Practices

- ❖ Start with a simple solution
- ❖ Test with various inputs
- ❖ Document your code
- ❖ Consider edge cases
- ❖ Optimize only when necessary

CoGrammar

Performance Considerations

- ❖ When designing algorithms, consider:
  - ➢ Time Complexity
  - ➢ Space Complexity
  - ➢ Input Size
  - ➢ Hardware Limitations
  - ➢ User Requirements

Review and Next Steps

- Key Takeaways:
  - Algorithms are step-by-step solutions to problems
  - Good algorithms are correct, efficient, and readable
  - Practice different problem-solving methodologies
- Next Steps:
  - Continue practicing
  - Explore real-world applications

Additional Resources

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

CoGrammar

Which of these methodologies works by breaking the problem into smaller parts?

A. Brute Force
B. Divide-and-Conquer
C. Greedy Algorithm
D. Randomised Approach

CoGrammar

Brute force algorithms are always the most efficient.

A. True
B. False

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education