




Welcome to the **Co**Grammar Recursion, Sorting and Searching (Tutorial)

The session will start shortly...

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



Software Engineering Session Housekeeping

- 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. Moderators are going to 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](#)

Software Engineering Session Housekeeping cont.

- 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](#)

Skills Bootcamp

8-Week Progression Overview

Fulfil 4 Criteria to Graduation

✓ Criterion 1: Initial Requirements

- **Guided Learning Hours (GLH):**
Minimum of 15 hours
- **Task Completion:** First 4 tasks

Due Date: 24 March 2024

✓ Criterion 2: Mid-Course Progress

- **Guided Learning Hours (GLH):**
Minimum of 60 hours
- **Task Completion:** First 13 tasks

Due Date: 28 April 2024

Skills Bootcamp Progression Overview

✓ Criterion 3: Course Progress

- **Completion:** All mandatory tasks, including Build Your Brand and resubmissions by study period end
- **Interview Invitation:** Within 4 weeks post-course
- **Guided Learning Hours:** Minimum of 112 hours by support end date (10.5 hours average, each week)

✓ Criterion 4: Demonstrating Employability

- **Final Job or Apprenticeship Outcome:** Document within 12 weeks post-graduation
- **Relevance:** Progression to employment or related opportunity

Learning Objectives & Outcomes

- Define **recursion** and identity a recursion problem
- Implement **recursion** for basic problems like factorial or binary search
- Predict **stackoverflow** from ill-formed recursion
- Understand basic **searching algorithms** and their associated complexities: **Linear and Binary Search**
- Understand based **sorting algorithms** and their associated complexities: **Bubble and insertion sort**
-

**SKILLS
FOR LIFE**
SKILLS BOOTCAMPS



Department
for Education

CoGrammar Recursion

April 2024

Recursion Poll

1. Which of the following best defines a base case in recursive functions?
 - a. The case where the function calls itself
 - b. The case that terminates the recursive calls
 - c. The case where the function returns a value
2. What is the maximum depth of recursion that can be achieved in most programming languages?
 - a. Limited by the size of the call stack
 - b. Unlimited, as modern compilers handle recursion efficiently
 - c. Limited by the size of the heap memory
3. When comparing recursive and iterative solutions for the same problem, what are some advantages and disadvantages of each approach?
 - a. Recursion typically uses less memory but may be slower.
 - b. Recursion can lead to more elegant and readable code, but iterative solutions are often more efficient in terms of speed and memory usage.
 - c. Recursion is always faster and more memory-efficient than iteration.

What are Recursion and Iterations?

- **Recursion** is a programming technique where a function calls itself to solve a problem by breaking it down into smaller, similar subproblems.
- **Iteration** is a fundamental programming concept that involves repeating a set of instructions or a process multiple times until a specific condition is met.

Types of iterations

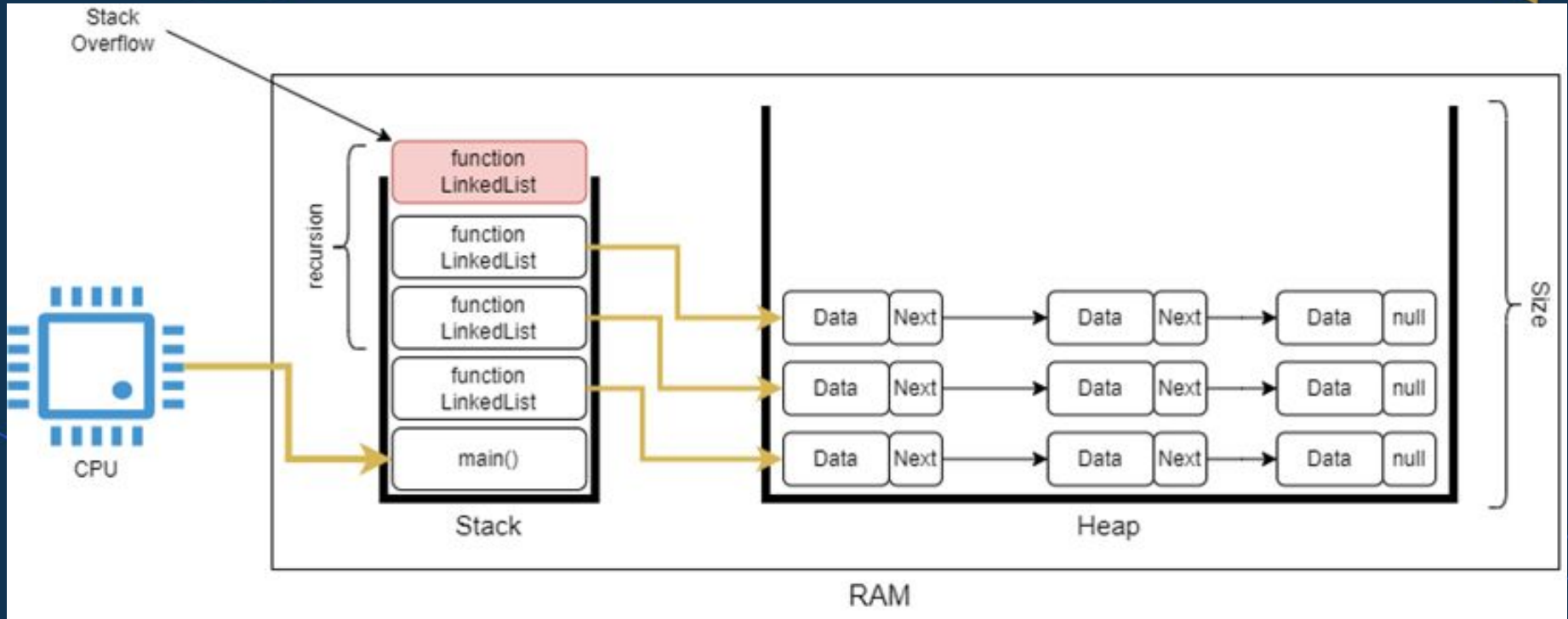
- **Count-controlled Iterations**
- **Sentinel-controlled Iteration**
- **Condition-controlled Iterations**

Iterations vs Recursions

In a **recursion**, a function calls itself until it reaches a **base condition**. It's often used for problems that can be broken down into similar sub-problems.

Iteration, on the other hand, repeats a set of instructions in a loop until a **condition is met**.

Stackoverflow



Let's get coding!



Questions and Answers



CoGrammar Sorting

April 2024

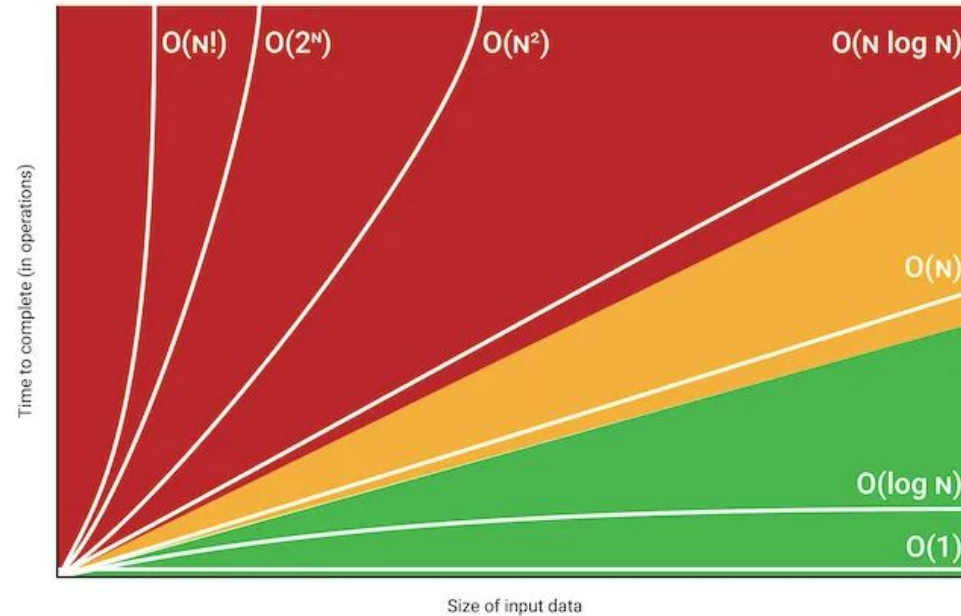
Sorting Poll

1. What is the time complexity of bubble sort?
 - a. $O(n^2)$
 - b. $O(n \log(n))$
 - c. $O(\log(n))$
2. What is the main advantage of merge sort over bubble sort?
 - a. Merge sort has a better time complexity ($O(n \log n)$)
 - b. Merge sort has a smaller memory footprint
 - c. Merge sort is easier to implement

Data Structures and Algorithms

- A **data structure** is a specialized format for organizing, processing, retrieving and storing data.
Eg: Tree, List, Stacks, Queues
- An **algorithm** is a set of commands that must be followed for a computer to perform calculations or other problem-solving operations.
Eg: Searching, Sorting

Order of Complexity



Sorting Algorithms Definition

A **Sorting Algorithm** is used to rearrange a given array or list of elements according to a comparison operator on the elements.

Sorting Algorithms

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order, continuing until the list is sorted.

Sorting Algorithms

Insertion sort is a sorting algorithm that builds the final sorted array one item at a time by repeatedly taking the next element and inserting it into the correct position in the already sorted part of the array.

Sorting Algorithms

Selection sort is a sorting algorithm that repeatedly selects the minimum element from the unsorted portion of the array and swaps it with the first unsorted element, gradually building up a sorted array from left to right.

Let's get coding!



Questions and Answers





**SKILLS
FOR LIFE**
SKILLS BOOTCAMPS



Department
for Education

CoGrammar Searching

April 2024

Searching Poll

1. **What is the significance of Big O notation in algorithm analysis, and how does it help in comparing the efficiency of different algorithms?**
 - a. Big O notation represents the maximum time complexity of an algorithm, and it helps in comparing the actual execution time of algorithms.
 - b. Big O notation provides an upper bound on the worst-case time complexity of an algorithm, allowing for a comparison of how the algorithm's performance scales with input size.
 - c. Big O notation denotes the average time complexity of an algorithm, facilitating a comparison of the expected performance of different algorithms.
2. **What is the fundamental difference between a stack and a queue, and how does each data structure prioritize element access?**
 - a. Stacks prioritize elements based on their priority level, while queues prioritize elements based on their insertion order.
 - b. Stacks follow the Last In, First Out (LIFO) principle, where the last element added is the first one to be removed, while queues follow the First In, First Out (FIFO) principle, where the first element added is the first one to be removed.
 - c. Stacks and queues both prioritize elements based on their alphabetical order.

Searching Poll

1. Explain the binary search algorithm and its key principles for efficiently searching for an element in a sorted array.
 - a. Binary search involves linearly scanning through each element of the array to find the target element.
 - b. Binary search divides the array into halves, compares the target element with the middle element, and recursively searches the appropriate half, reducing the search space by half with each iteration until the target element is found or the search space is empty.
 - c. Binary search involves sorting the array first and then searching for the target element using a linear search algorithm.

Searching Algorithms Definition

Searching algorithms are essential tools in computer science used to locate specific items within a collection of data.

Searching Algorithms

Linear search is a simple search algorithm that sequentially checks each element in a list until the target element is found or the end of the list is reached.

Searching Algorithms

Binary search is a search algorithm that efficiently locates a target value within a sorted array by repeatedly dividing the search interval in half and comparing the target value to the middle element, eliminating half of the remaining elements each time.

Let's get coding!



Questions and Answers



Thank you for attending



Department
for Education

CoGrammar

