

SEARCHING AND SORTING

DR IAN CORNELIUS

HELLO

- Learning Objectives
 1. Understand the concept of searching and sorting in Python
 2. Demonstrate the ability to use sorting algorithms

INTRODUCTION TO SEARCHING ALGORITHMS

- Searching algorithms are a series of instructions to retrieve information stored in a data structure
- Split into two categories:
 1. Sequential
 2. Interval
- There are many searching algorithms
 - i.e. linear search, binary search, interpolation search and exponential search

LINEAR SEARCH

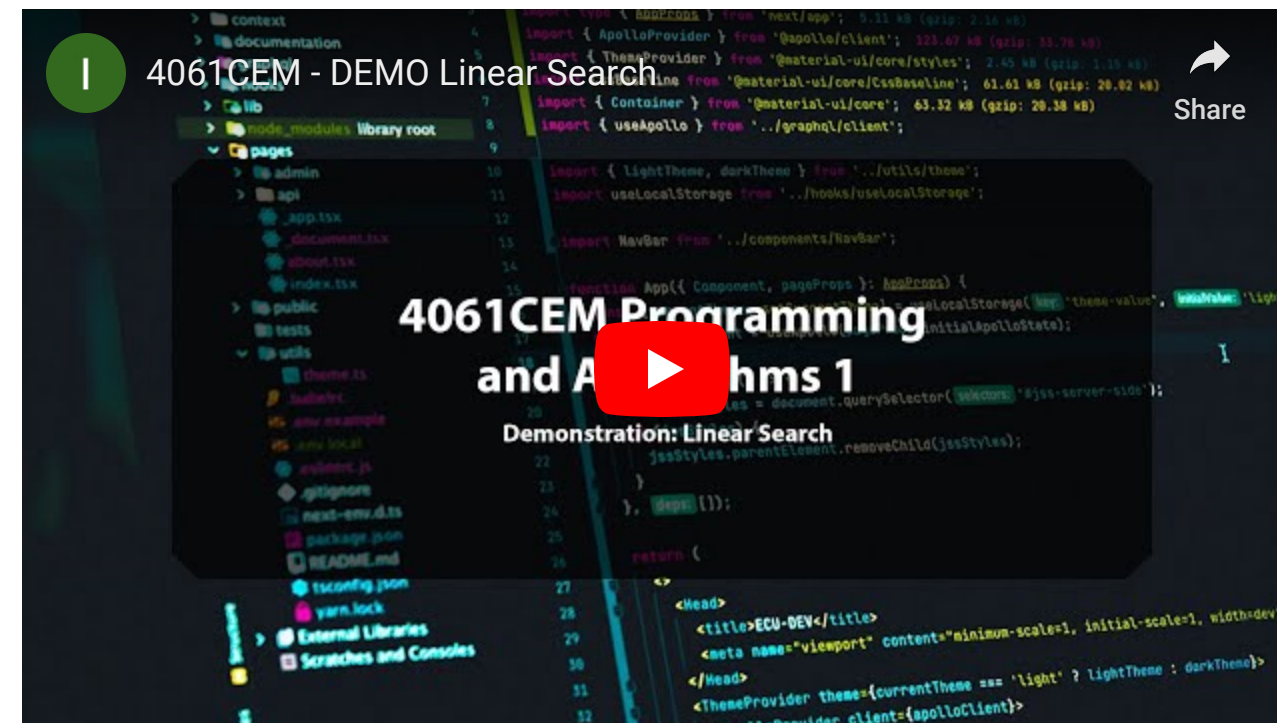
- A method to find a target value in a list
- It will sequentially check each element of the list for a target value
 - this is repeated until a match is found or until all elements have been searched

ALGORITHM INSTRUCTIONS

1. Start at the left-most (or right-most) element of the list
2. One-by-one compare each element of the array to a target
3. If the element matches the target, return the index number
4. If the element does not match the target, go back to step 2
5. If any of the elements do not match the target value, then return `False`

LINEAR SEARCH DEMONSTRATION

- Demonstration of the Linear Search algorithm
 - Refer to the pre-recorded video for a demonstration



BINARY SEARCH

- An efficient search algorithm to solve problems
- It will search a sorted array by dividing the search interval in half

ALGORITHM INSTRUCTIONS

1. Compare the target value with the middle element
2. If the target value matches with the middle element, return the middle index
3. If the target value is greater than the middle element, perform a search on the right half of the list
4. If the target value is less than the middle element, perform a search on the left half of the list
5. If the element does not match any of the elements, return `False`

BINARY SEARCH DEMONSTRATION

- Demonstration of the Binary Search algorithm
 - Refer to the pre-recorded video for a demonstration



INTRODUCTION TO SORTING

- Sorting algorithms are made up of a series of instructions
 - they accept a list as an input and outputs a sorted list
- There are many examples of sorting algorithms:
 - i.e. selection sort, bubble sort, insertion sort, merge sort etc.
- In this lecture we shall look at two sorting algorithms: bubble and insertion sort

BUBBLE SORT

- A simple algorithm that works by swapping adjacent elements if they are in the incorrect order

ALGORITHM INSTRUCTIONS

1. Compare each adjacent element in the list
2. Swap the two elements if necessary
3. Repeat the process for all elements in the list until the list is sorted

BUBBLE SORT DEMONSTRATION

- Demonstration of the Bubble Sort algorithm
 - Refer to the pre-recorded video for a demonstration



INSERTION SORT

- A sorting algorithm that works in a similar method to how we would sort playing cards in our hands

ALGORITHM INSTRUCTIONS

1. Compare the first element (n) of the list to the next one ($n + 1$)
2. If n is less than $n + 1$, add n to the sorted sub-list, move onto the next element
3. Compare the current element to all the elements in the sub-list
4. Shift all elements in the sorted sub-list if it is greater than the current element
5. Insert the current element into the sorted sub-list
6. Repeat this process until the list is sorted

INSERTION SORT DEMONSTRATION

- Demonstration of the Insertion Sort algorithm
 - Refer to the pre-recorded video for a demonstration



GOODBYE

- Questions?
 - Post them in the **Community Page** on Aula
- Contact Details:
 - Dr Ian Cornelius, ab6459@coventry.ac.uk