

Data Structures in JavaScript.

Niklaus Wirth, a Swiss computer scientist, wrote a book in 1976 titled *Algorithms + Data Structures = Programs*. 40+ years later, that equation still holds true. That's why software engineering candidates have to demonstrate their understanding of data structures along with their applications.

Almost all problems require the candidate to demonstrate a deep understanding of data structures. It doesn't matter whether you have just graduated (from a university or coding boot camp), or you have decades of experience. In our fifth week of our 20 weeks coding bootcamp will be all about Data Structures in JavaScript.

Introduction to Data structures.

21st Day

- What is a Data Structure?
- Why do we need Data Structures?
- Commonly used Data Structures.
 - a) Arrays.
 - b). Stacks.
 - c). Queues.
 - d). Linked Lists.
 - e). Trees.
 - tries
 - f). Graphs.
 - g). Hash Tables.

An array is the simplest and most widely used data structure. Other data structures like stacks and queues are derived from arrays.

Basic Operations on Arrays to be discussed:

- 1). Insert — Inserts an element at a given index
- 2). Get — Returns the element at a given index
- 3). Delete — Deletes an element at a given index
- 4). Size — Gets the total number of elements in an array

At the end of the day learner should be able to:

- 1). Find the second minimum element of an array
- 2). First non-repeating integers in an array
- 3). Merge two sorted arrays
- 4). Rearrange positive and negative values in an array

Basic operations of stack to be discussed:

- 1). Push — Inserts an element at the top
- 2). Pop — Returns the top element after removing from the stack
- 3). isEmpty — Returns true if the stack is empty
- 4). Top — Returns the top element without removing from the stack

At the end of the day learner should be able to:

- 1). Evaluate postfix expression using a stack
- 2). Sort values in a stack
- 3). Check balanced parentheses in an expression

Basic operations of Queue to be discussed:

- 1). Enqueue() — Inserts an element to the end of the queue
- 2). Dequeue() — Removes an element from the start of the queue
- 3). isEmpty() — Returns true if the queue is empty
- 4). Top() — Returns the first element of the queue

At the end of the day learner should be able to:

- 1). Implement stack using a queue
- 2). Reverse first k elements of a queue
- 3). Generate binary numbers from 1 to n using a queue

Types of linked lists:

- a). Singly Linked List (Unidirectional)
- b). Doubly Linked List (Bi-directional)

Basic operations of Linked List:

- 1). InsertAtEnd — Inserts a given element at the end of the linked list
- 2). InsertAtHead — Inserts a given element at the start/head of the linked list
- 3). Delete — Deletes a given element from the linked list
- 4). DeleteAtHead — Deletes the first element of the linked list
- 5). Search — Returns the given element from a linked list
- 6). isEmpty — Returns true if the linked list is empty

At the end of the day learner should be able to:

- 1). Reverse a linked list
- 2). Detect loop in a linked list
- 3). Return Nth node from the end in a linked list.
- 4). Remove duplicates from a linked list

So far we will have covered the linear Data Structures and we will be in a position to start on Frameworks.

On 15th August 2020 we will have a session on “introduction to Angular Framework” By Wayne Gakuo, Software Engineer @ MySpace property.

We will later cover the basic non-linear Data Structures: Graphs, Trees and Hash Tables.

Lux Tech Academy.

Email: luxtinc@gmail.com

call : +254794598834

Twitter: @lux_academy or @DataScience_Ea