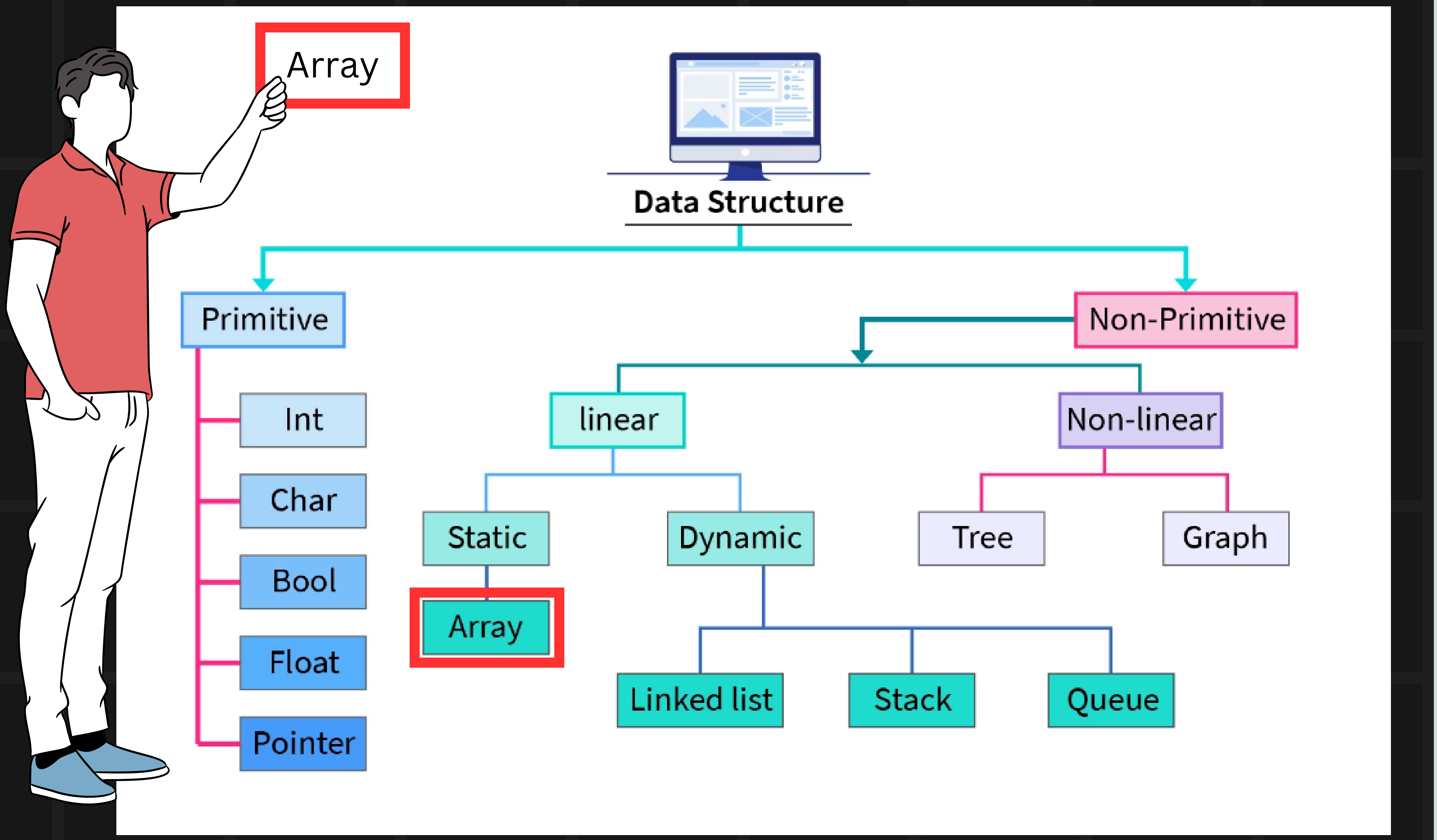
Data Structure:



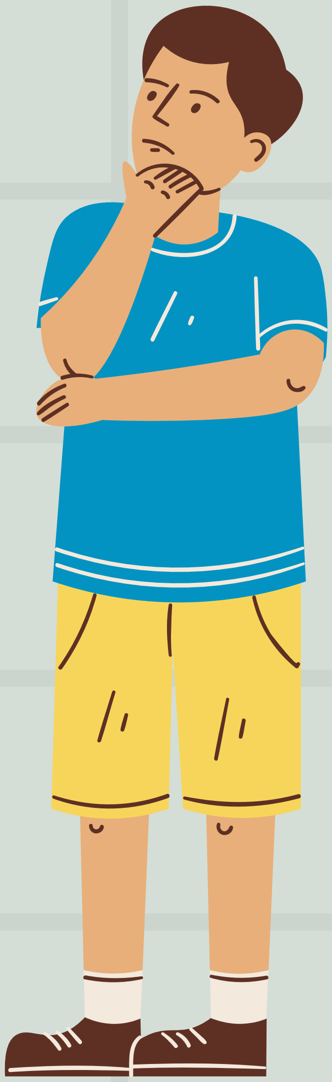
A DATA STRUCTURE IS LIKE A CONTAINER THAT HELPS YOU ORGANIZE AND STORE DATA IN A WAY THAT MAKES IT EASY TO USE AND MANAGE.

Arrays:

01

Operations :

1. Insertion
2. Deletion
3. Traversal
4. Search
5. Updating
6. Sorting
7. Merging
8. Splitting
9. Reversing



AN ARRAY IS A COLLECTION OF ITEMS STORED TOGETHER IN A SINGLE VARIABLE.

Types of Arrays :

1. Single-Dimensional Array

-simple linear array that holds elements in a single row.

2. Multi-Dimensional Array

-Arrays that can hold elements in multiple dimensions (like rows and columns).

3. Dynamic Array

-An array that can change in size during runtime. Unlike static arrays, dynamic arrays can grow or shrink as needed.

4. Associative Array/Hash Array

-An array that uses key-value pairs instead of numerical indices. This allows for more flexible data retrieval.

5. Sparse Array:

-An array in which most elements are zero or empty. Sparse arrays can be optimized to save memory.

Types of Array sorting :

03

1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Merge Sort
5. Quick Sort
6. Heap Sort
7. Counting Sort
8. Radix Sort
9. Bucket Sort

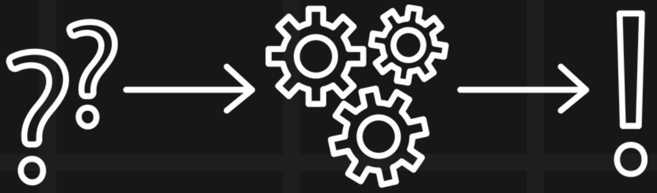
SORTING IS THE PROCESS OF ARRANGING THE ELEMENTS OF A COLLECTION IN A SPECIFIC ORDER.THIS ORDER CAN BE EITHER ASCENDING OR DESCENDING.

1. Bubble Sort :

BUBBLE SORT REPEATEDLY STEPS THROUGH THE LIST, COMPARES ADJACENT ELEMENTS, AND SWAPS THEM IF THEY ARE IN THE WRONG ORDER. THE PROCESS IS REPEATED UNTIL NO SWAPS ARE NEEDED, INDICATING THAT THE LIST IS SORTED.

2. Selection Sort :

SELECTION SORT DIVIDES THE ARRAY INTO A SORTED AND AN UNSORTED SECTION. IT REPEATEDLY SELECTS THE SMALLEST (OR LARGEST) ELEMENT FROM THE UNSORTED SECTION AND MOVES IT TO THE END OF THE SORTED SECTION.



3. Insertion Sort :

INSERTION SORT BUILDS A SORTED ARRAY ONE ELEMENT AT A TIME. IT TAKES EACH ELEMENT FROM THE UNSORTED PORTION AND INSERTS IT INTO THE CORRECT POSITION IN THE SORTED PORTION.

4. Merge Sort :

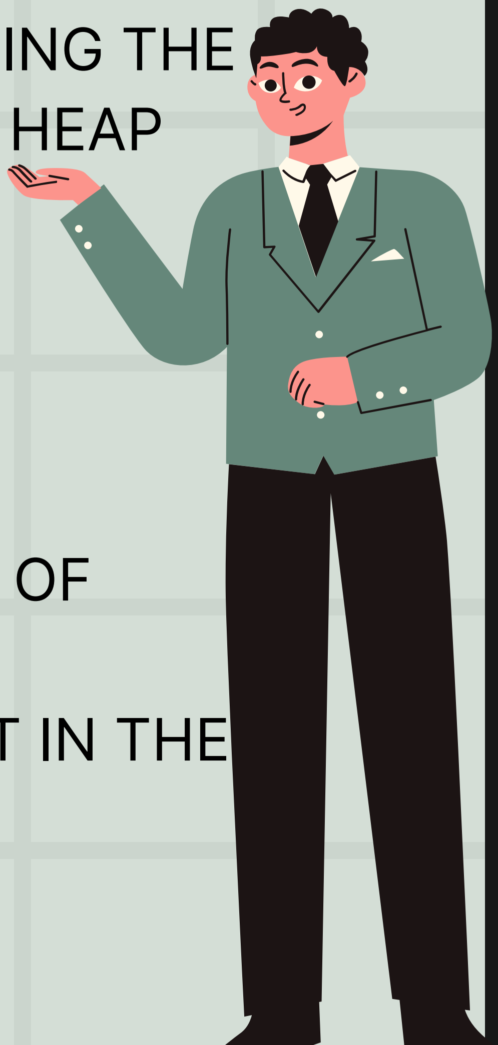
MERGE SORT IS A DIVIDE-AND-CONQUER ALGORITHM. IT DIVIDES THE ARRAY INTO TWO HALVES, RECURSIVELY SORTS EACH HALF, AND THEN MERGES THE SORTED HALVES BACK TOGETHER.

5. Quick Sort :

QUICK SORT PICKS A "PIVOT" ELEMENT FROM THE ARRAY AND PARTITIONS THE OTHER ELEMENTS INTO TWO GROUPS: THOSE LESS THAN THE PIVOT AND THOSE GREATER. IT THEN RECURSIVELY SORTS THE SUBARRAYS.A PIVOT IS AN ELEMENT SELECTED FROM THE ARRAY THAT IS USED TO PARTITION THE OTHER ELEMENTS INTO TWO GROUPS

6. Heap Sort :

HEAP SORT INVOLVES BUILDING A BINARY HEAP FROM THE INPUT DATA, THEN REPEATEDLY EXTRACTING THE MAXIMUM (OR MINIMUM) ELEMENT FROM THE HEAP AND RECONSTRUCTING THE HEAP UNTIL ALL ELEMENTS ARE SORTED.



7. Counting Sort :

COUNTING SORT COUNTS THE OCCURRENCES OF EACH UNIQUE ELEMENT IN THE ARRAY, THEN CALCULATES THE POSITION OF EACH ELEMENT IN THE SORTED ARRAY BASED ON THESE COUNTS.

8. Radix Sort :

COUNTING SORT COUNTS THE OCCURRENCES OF EACH UNIQUE ELEMENT IN THE ARRAY, THEN CALCULATES THE POSITION OF EACH ELEMENT IN THE SORTED ARRAY BASED ON THESE COUNTS.

9. Bucket Sort :

BUCKET SORT DIVIDES THE INPUT ARRAY INTO SEVERAL "BUCKETS." EACH BUCKET IS THEN SORTED INDIVIDUALLY (USING ANOTHER SORTING ALGORITHM) AND THE SORTED BUCKETS ARE COMBINED TO FORM THE FINAL SORTED ARRAY.