Data Structure and Algorithms Linked lists, singly, doubly, circular? kruskal method, prims method, ..? hashing? max heap, min heap? binary search? merge sort,quick sort,heap sort,insertion sort, selection sort? binary tree, binary search tree? linear probing? ipv4,ipv6

What is dsa?

In simple language, Data Structures are structures programmed to store ordered data, so that various operations can be performed on it easily. It represents the knowledge of data to be organized in memory. It should be designed and implemented in such a way that it reduces the complexity and increases the efficiency.

Store a data into organized way so that required functions can be perform dsa isused to store large and connected data.

**Primitive:**

The **primitive** data types include byte, int, long, short, float, double, and char. They are part of the core of Java and you don't need anything special to use them. For example, the following declares a long variable for a partNumber.

A data type that is primitive, such as the long variable, actually stores the value. If we give a value to the partNumber value, for example 4030023, that is what Java stores.

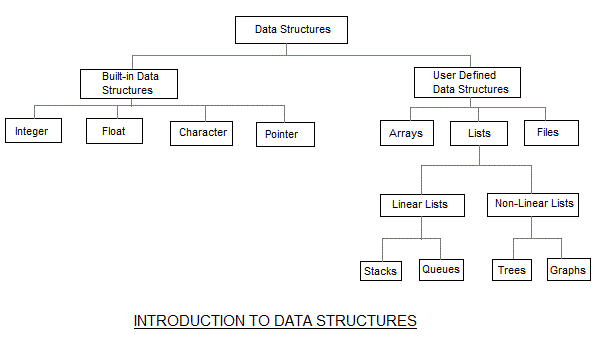
|  |
| --- |
| Java primitive data declaration |

**Non-primitive:**

**It** store the refereance or address of memory where value is stored.

* **Non-primitive**, or reference data types, are the more sophisticated members of the data type family. They don't store the value, but store a reference to that value. Instead of partNumber 4030023, Java keeps the reference, also called address, to that value, not the value itself. **String**,**array**,Enum,Class,etc
* Linked List
* Tree
* Graph
* Stack, Queue etc.

All these data structures allow us to perform different operations on data. We select these data structures based on which type of operation is required.



The data structures can also be classified on the basis of the following characteristics:

|  |  |
| --- | --- |
| **Characterstic** | **Description** |
| Linear | In Linear data structures,the data items are arranged in a linear sequence. Example: **Array** |
| Non-Linear | In Non-Linear data structures,the data items are not in sequence. Example: **Tree**, **Graph** |
| Homogeneous | In homogeneous data structures,all the elements are of same type. Example: **Array** |
| Non-Homogeneous | In Non-Homogeneous data structure, the elements may or may not be of the same type. Example: **Structures** |
| Static | Static data structures are those whose sizes and structures associated memory locations are fixed, at compile time. Example: **Array** |
| Dynamic | Dynamic structures are those which expands or shrinks depending upon the program need and its execution. Also, their associated memory locations changes. Example: **Linked List created using pointers** |

***Space complexity****is the amount of memory used by the algorithm (including the input values to the algorithm) to execute and produce the result*

***Space complexity*** Time complexity of an algorithm signifies the total time required by the program to run till its completion