

Data Structures

Lecture Details:
Data Structures

Introduction



- Data structure is a representation of logical relationship existing between individual elements of data.
- In other words, a data structure defines a way of organizing all data items that considers not only the elements stored but also their relationship to each other.
- The term data structure is used to describe the way data is stored.
- To develop a program of an algorithm we should select an appropriate data structure for that algorithm.
- Therefore, data structure is represented as:

Algorithm + Data structure = Program

Introduction

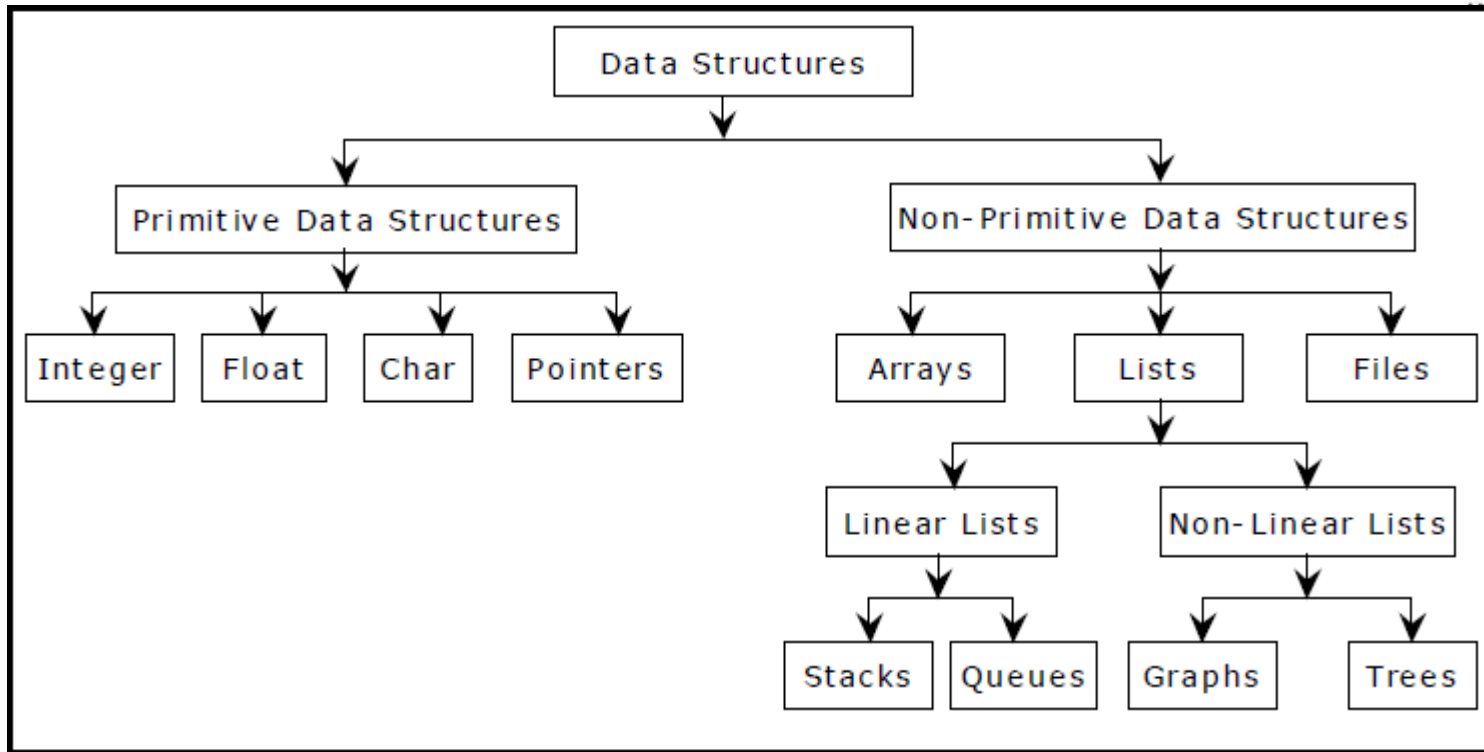


- A data structure is said to be linear if its elements form a sequence or a linear list.
- The linear data structures like an array, stacks, queues and linked lists organize data in linear order.
- A data structure is said to be non linear if its elements form a hierarchical classification where, data items appear at various levels.
- Trees and Graphs are widely used non-linear data structures.
- Tree and graph structures represents hierarchical relationship between individual data elements.
- Graphs are nothing but trees with certain restrictions removed.

Types

- **Primitive Data Structures** are the basic data structures that directly operate upon the machine instructions.
- They have different representations on different computers. Integers, floating point numbers, character constants, string constants and pointers come under this category.
- **Non-primitive data structures** are more complicated data structures and are derived from primitive data structures.
- They emphasize on grouping same or different data items with relationship between each data item. Arrays, lists and files come under this category.

Types



Organization of Data



- The collection of data you work with in a program have some kind of structure or organization.
- No matter how complex your data structures are they can be broken down into two fundamental types:
 - Contiguous
 - Non-Contiguous.

Contiguous Vs Non Contiguous

- In contiguous structures, terms of data are kept together in memory (either RAM or in a file).
- An array is an example of a contiguous structure. Since each element in the array is located next to one or two other elements.
- In contrast, items in a non-contiguous structure are scattered in memory, but are linked to each other in some way.
- A linked list is an example of a non-contiguous data structure.

Contiguous Vs Non Contiguous

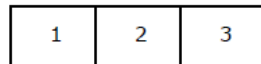


(a) Contiguous



(b) non-contiguous

```
int arr[3] = {1, 2, 3};
```



(a) Array

```
struct cust_data
```

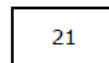
```
{
```

```
    int age;
```

```
    char name[20];
```

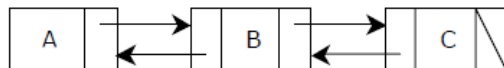
```
};
```

```
cust_data bill = {21, "bill the student"};
```

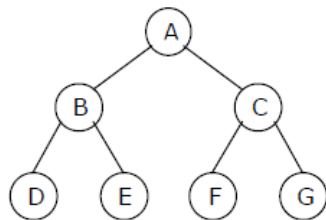


(b) struct

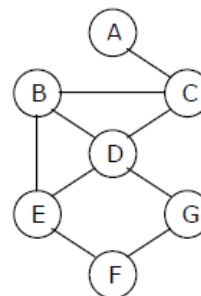
"bill the student"



(a) Linked List



(b) Tree



(c) graph

Thank You