Abstract data type

Abstract Data type is a class of entities with a set of values and a set of operations describing their behaviour. Abstract types of data are mathematical models of a set of data values or information that share similar behaviour or attributes and can be defined and described irrespective of actual implementation. And Usually, abstract forms of data are used in algorithms. The abstract data type is specified instead of being implemented in terms of its data items or its related operations. And Abstract data types, abstract data structures or forms are specified indirectly only by the operations that can be performed on them and by mathematical constraints on their results.

ADT are the data types which are made up of or composed of these primitive data types as these ADT can be application or implementation specific, they could be created on need basis.

Here are some examples

* **stack:** operations are "push an item onto the stack", "pop an item from the stack", "ask if the stack is empty"; implementation may be as array, linked list,
* **queue:** operations are "add to the end of the queue", "delete from the beginning of the queue", "ask if the queue is empty"; implementation may be as array or linked list

ADT is to an interface, what a data structure is to a class.

A few examples:

**ADT** – List

**DS**: Array List, Linked List, Vector List

**ADT** – Map

**DS:** Hash Map, Tree Map, Sorted Map

**ADT** – Set

**DS** –Hash Set, Tree Set, Linked Hash Set

The fundamental difference between the abstract data type (ADT) and the concrete data type is that the latter helps us to look at the actual representation, while the latter hides from us the representation. An ADT may be pure ADT or Updatable ADT. One where all operations are pure functions is a pure ADT. It means that there are no side effects of operations. In addition, input arguments are not changed or revised there. We simply use these arguments to produce output, which are ADT (or other forms of fresh values). Many forms of concrete are simple. For example, no integers operation actually modifies an integer. Alternatively, fresh production is provided by all operations like' +.'

Data Structure

A data structure is a specific way of organizing information in a system for efficient use. as an example, using the array data structure, we can store a list of items with the same data type.

|  |  |  |  |
| --- | --- | --- | --- |
| U | B | F | D |

0 1 2 3

I have listed below popular data structures

* array
* Stack
* Queue
* Binary Tree
* Graph
* Heap
* Metrics
* Linked List

Queue Data Structure

A Queue is a linear structure that follows a specific order performing the operations. First in First Out (FIFO) is ordered. A good example of a queue is any customer queue for a service where the first-coming user is first served. The difference is in reducing stacks and queues. In a stack we delete the newly added element; in a queue we remove the newest added object.

Sorting algorithms