Introduction to Data Structures and Algorithms

Computers are used to Solve Problems

3 Steps to solve a problem on a Computer :

- A **model** of the problem
- An **algorithm** within the framework of the model
- Computer representation of **data**

Knowledge of different **data models** and relevant **operations/algorithms** is essential for **objective** solution of problem.

Models

- What is a System?
  - A system is an interconnection of components or subsystems
- What is a Model?
  - A Model is a Collection of Properties of the target system
- Why are Models required?
  - To develop Intuition and Knowledge
- Approaches to develop
  - Experiments and Observations
  - Modeling and Analysis
- Models should be Accurate enough and Tractable

- The World has no responsibility to follow the Model made by Humans

Information & Data

- **Information**

  Some fact about the surrounding

  e.g.

  Height of Kamal is 6.5 ft.

  Rainfall today was 10 mm.

Information & Data …

- **Data**

  Abstraction of information in a Problem Solving System

e.g.

6.5

10

Data may be numeric or non-numeric.

Of course, at the machine level all kinds of data are represented as strings of bits : 1 and 0.

Data and Operations

- For representation of different forms of data, different data types are used.

- Each data type contains a set of allowable values and a set of allowable operations.

- Data values are interpreted according to their types.

  e.g. 123 – an integer

  "123" – a string of characters

- Operations also depend on the corresponding data types.

  e.g. 123 + 45 = 168 – integer addition

  "123"+"45" = "12345" – string concatenation

Program Structures

- Program = Algorithm + Data Structure

- Programming Languages provide facilities for algorithm representation and data representation.

- High Level Programming Languages like PASCAL and C facilitate structured and modular programming by providing algorithm structures and sub-program structures

- Algorithm structures are :

1. Sequence
2. Conditional
3. Iteration
4. Subprogram
5. Recursion
6. Control transfer

Data Types

- **Scalar**

  Integer Real Character Boolean

  Pointer Subrange Enumerated

- **Data Aggregation Facilities**

  Arrays Records Sets

- Structured data types
- (1) **Components**
- (2) **Structure** defined by the set of rules that put the components together
- (3) **Set of operations**

ABSTRACT DATA TYPE (ADT)

- A **conceptual model** of information structure.
- An ADT **specifies** the **components**, their **structuring relationships** and a list of operations that are allowed to be performed.
- ADTs are independent of data representation and implementation of operations
- It is just a **specification**, no design or implementation info is included.
- The components themselves are other ADT’s.

ADT …

- No assumption is made about the range of values of the components.
- Specification involves the “**what**”s of the operations, not the “**how**”s.
- ADT’s are generalizations of **primitive** data types.
- They **encapsulate** data values.

Generic Operations on ADT

- Create and Dispose a Data Abstraction
- Insert data element/component into the Data Abstraction as per structural relationship
- Access/Search a data element/component in the populated Data Abstraction
- Remove/Delete a data element/component from the populated Data Abstraction
- The above operations are structural in nature. In addition, there may be some domain-dependent operations also.

Data Structure

- A data structure is the **design representation** of an ADT.

- The same ADT may be represented by several data structures.

- Eg :

  Real nos : (1) <int> . <int>

  (2) (<int> , <int>)

Data Structure …

- There are many data structures corresponding to the ADT “set”.

- Operations on data structures are represented as algorithms.

- If the relations among components are not implicitly known,

  (1) they can be expressed separately (possibly through another data structure)

  (2) Components can be augmented to include additional fields that represent structural information. These components are termed as nodes.

Data Structure …

- A data structure is implemented using the facilities of a programming language.

- So what we use as a data structure is an ADT restricted at the levels of design and implementation.

Types of Data Structure

- Data Structures are of two generic types:

  Static and Dynamic

- The memory requirement of static data structures is fixed at the compile time.
- Dynamic data structures can shrink and grow at the run-time. Thus they require special memory allocation and deallocation functions from the programming system.
- **Algorithms and memory usage** of data structures should be as efficient as possible. Why?

Summary

- Data Types and Structured Data Types discussed
- Algorithmic Structures discussed
- Abstract Data Type, Data Structure and its Implementation defined and exemplified
- Static and Dynamic Data Structures discussed

THANK YOU