Data Structure & Algorithm Analysis

Biswajit Prasad
Assistant Professor
Department of Computer Science
Maharaja Manindra Chandra College
Calcutta 700 003

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.

Information & Data

- **Information**

  Some fact about the surrounding
  e.g.

  Height of Kamal is 6.5 ft. Rainfall today was

  10 mm.

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 facilitates structured and modular programming by providing algorithm structures.

- Algorithm structures are :

  1 . Sequence 2. Conditional

  3 . Iteration 4. Subprogram

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.
- 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.

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>)

Array as a Data Structure

ADT array

- Objects Elements of the same type arranged in a sequence. An associated index has finite ordinal type. There is an one-to-one correspondence between the values of the index and the array elements.

- Operations
- (1) store_array (a,i,e) -- store e's value in the ith element of array a
- (2) retrieve_array (a,i) -> e -- return the value of the ith element of array a

Array as a Data Structure

ADT array

- Objects Elements of the same type arranged in a sequence. An associated index has finite ordinal type. There is an one-to-one correspondence between the values of the index and the array elements.

- Operations
- (1) store_array (a,i,e) -- store e's value in the ith element of array a
- (2) retrieve_array (a,i) -> e -- return the value of the ith element of array a

Array as a Data Structure…

- **Design**

  The required no. of memory locations are statically allocated consecutively.

- **Implementation**

  Built into the language. What are the constraints ?

Polynomials – Application of Array

- Operations
  - Is-zero – returns true if polynomial is zero.
  - Coef – returns the coeff. of a specified exponent.
  - add - add two polynomials
  - mult - multiply two polynomials
  - Cmult - multiply a polynomial by a const.
  - attach – attach a term to a polynomial
  - remove – remove a term from a polynomial
  - degree – returns the degree of the polynomial

- <u>Representation decisions</u>
- 1. Exponents should be unique and be arranged in decreasing order.
- 2. Storage alternatives ?