Philosophy of Data Structures in Engineering Education

Jayalakshmi G Naragund*, Prakash A Kotre†, Suvarna Kanakaraddi‡, Sujata C§
* ‡ § Email: { jaya_gn, suvarna_gk, sujata_c }@bvb.edu, B V B College of Engg. & Tech., Hubli, India,
† Email { prakash.kotre@gmail.com} Samsung Bengaluru, India

***Abstract*—The pillars of Computer Science and Engineering (CSE) curriculum are Data Structures, Database management systems, languages, operating systems and algorithms. This article explains the relationship and connectivity of these core courses. Authors follow the pedagogy technique to teach the Data Structures (DS) by considering its evolution. The article focus on the method of providing connectivity between the building blocks of DS like Data Containers, Container Iterators, Algorithms and Functors. Each data structure is explained by considering its property, iterations, problems and applications. The three fold method is followed to teach DS, which includes think, build and discuss phases. The pedagogy techniques practiced by authors are revealed many mysteries, which are not discussed in most of the DS text books.**

***Index Terms*—Building blocks of Data Structures, Analysis of Data structures, Pedagogy, Problem Solving and Teaching aids.**

I. Introduction

Engineering education in the nation is the center for develop such intellectually trained engineers. Curriculum structure drives the education and decides the level of competence in the students. Curriculum design is the dynamic process in the technical education; it considers the industry demands and surrounding environmental changes.

Data structures core course includes the frame work to store and access data to and from computer. Beginners or exports use data structures to develop simple or complex software. All university syllabus and text book of data structures emphasis on the process of storing and retrieving the data but not on characteristics, correlation between the data structures. In this article authors contribute to describe the correlation between core courses of CSE and evolution, characteristics & correlation of data structures.

II. Pillars of Computer Science and Engineering

In these days computing field is the integral part each every discipline of engineering branches, hence curriculum design of CSE is playing vital role in amalgamation of computing with others. ACM (the Association for Computing Machinery) and the IEEE Computer Society are provided the guidelines for global computing field curriculum development.

Fig. 1 depicts the proposed core courses and their relationship. The outer circle of Fig. 1 connects pillars of core courses of CSE, which are Programming Languages, System Fundamental, Information Management and Software Development Fundamentals. Other core courses like Software Engineering, Networking & Communication and Architecture and Organization etc. belong to one of these pillars. Inner circle in Fig. 1 includes the Algorithms and Data structures core courses and these are the centric part of all other courses. In other words Algorithms and DS are implicitly included in all core courses and hence it is resided in inner circle of Fig. 1.

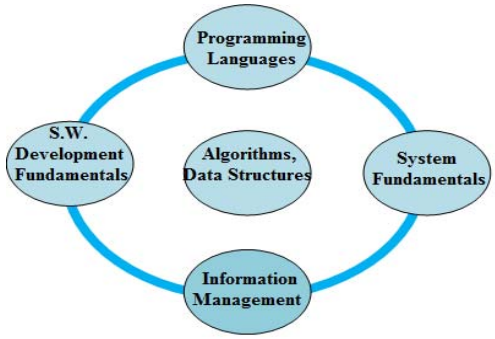

Fig. 1. Core courses of CSE and their relation

Programming Languages: Structured languages like C is used to develop Data Structures and Interfacing modules. So this becomes basic language across all engineering branches. Object oriented languages are fundamental of all internet web applications. Hence curriculum design must include at least one language from each group.

System fundamentals: Useful to develop knowledge of different system components, such as Computer organization, assembler, compiler design and software engineering etc.

Information management: Includes the courses, which are describing the organization or management of data.

Software Development Fundamentals: Web application development is big area where the most of customers of the S.W. industries belongs to this group. Technologies advancement is very fast in this area, so students should introduce the recent tool that is in demand, like PHP, HTML5 etc.

III. Philosophy of Data Structure

The Abstract data type of data structures illustrates logical storage structure and the operations those can be performed on them. The big picture of DS depicts that varies data structures are created by changing their properties.

A. Building Blocks of Data Structures

The main principle entities of DS are Data Containers, Container Iterators, Algorithms and Functors.

978-1-5090-2504-6/16 $31.00 © 2016 IEEE
DOI 10.1109/LaTiCE.2016.18

TABLE I
BASIC TYPE DATA STRUCTURES

Data structures	Properties	Iterations	Problems	Applications
Basic type	Values, Endianness (Big,Little)	bitwise operators: !, ∼ , &, |, ≪, ≫ etc.	small in size	Test a number for - Power of 2, Sign of numbers, Bit counting, Equivalence operation
Arrays	Capacity, Size, Order, Position, Majority (Row/column)	Loop strucures : for,while loop	Crashes ,Overflows, Contiguous requirements, know the size upfront	Sum/Avg/Min/Max, Minima & maxima (Linear, Optimal), Vectors maths, Matrix maths
Strings	Capacity, Size (strlen), Order, Position, Majority (Row/column), End marking, Case (small/big)	Fixed: Literals Variable, Length(java), Delimited(c/c++)	Forgetting delimits, Crashes, Overflows	Palindrome, Reverse, Comparison, Hash, (md5), Encryption.
Trees	Height,Non linear, variable size, 1-n relation	Tree traversal Insert, search, delete,	Dangling references	Dictionary, searching and sorting
Lists	linear, variable size, 1-1 relation	Singly ,Doubly, Circular, Insert, search, delete	Dangling references	Dictionary, Addition of long integers, addition of two polynomials
Graphs	Non linear, adjacency matrix, adjacency lists ,cyclic , Acyclic, Directed ,Weighted	Graph traversal	Improper connectivity, negative weights	Transportation problem, routing, scheduling

Data Containers: This entity provides the abstract view of data storage. Fig. 2. shows the evolutionary study of data containers. The progressive study begins from basic type structures, contiguous structures and ends with non-contiguous structures. All primitive data types of programming languages belong to this category.

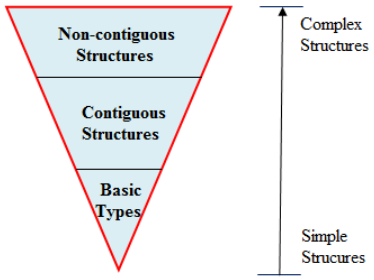

Fig. 2. Evolution of Data Structures

Container Iterators : Allows the programmer to traverse through the data structures. These are fixed to the DS and provide access to data items in the containers. It is important to know that these are not carrying out the iterations on DS.

Algorithm: Specifies the operations that can perform on the data structures. It exhibits the behavior of data structures.

Functors : Used in object oriented languages and objects behaves like function is called as functors, for example if instance of class behaves like function , then it is a functor.

B. Illustration of Data Structures

Authors used four parameters like property, iterations, problems and applications to explain the DS as shown in Table I . The properties disclose the features or attributes of data structures, iterations show the accessing of items, problems provides the pitfalls or limitations of DS and applications are service accepters.

IV. PEDAGOGY ACTIVITY ANALYSIS

The three fold pedagogy technique is used to teach DS of CSE students. The phases involved in the technique are think, build and discuss. Authors consider the sorting algorithm to illustrate the pedagogy technique .

Think phase : Students are motivated to think about the suitable type, properties and application of DS. It encourages the learners to know about all existing DS and answer the following questions:

- Which data structure's properties are satisfying the customer's or application requirements? (DS is array and properties are Min, Max, order and position)
- What type of building block is required? (Container)
- What are its applications? (Where order of items is important , Search engines, Indexing in DB)

Build phase: Students think about assumption, constraints and strategic plan of development of given application and. The build phase persuades the learners to reply the following questions:

- What conditions will become assumptions and constraints? (Assumption :static implementation , Constraint : Capacity)
- What is the relationship between DS and application? (Indexing and ordering)
- What all the operations of DS and suitable design technique are needed? (operation: finding Min and partition. Technique: Divide and conquer)

Discuss Phase: The focus of this phase is the comparative study of selected algorithms or analysis of an algorithm in Build phase. It requires the problem solving skill in the students to analyse the algorithms and force them to answer the following questions:

- What is mathematical model? (Linear)
- What is the time complexity class? (O(n log(n))
- Dose the selected algorithm perform better? (Yes)

V. CONCLUSION

The philosophy of DS explained in this article gives new dimensions to teach DS by using the properties, problems and applications of DS. The pedagogy technique used by authors makes the students more creative and implementation of application more challenging. The proposed techniques can be extended to other IT branches also.