

The Role of Data Structures in Multiple Disciplines of Computer Science- A Review

N. Shobha Rani, Suman S.P.

Abstract— This paper revises the applications of data structures associated with field of computer science and also with great relevance in accomplishing the functions of Operating system and improving the throughput by resource utilization. There is an amount of essential complexity in organizing/processing the data and files present in system. No brute force approaches can minimize and manifest it so easily by adopting to the non-synchronized methodology of storage and management, but it can be placed so that it doesn't get in the way and it's also easy to understand. The best such place is a data structure. Data structures have many applications in the area of system development, data base design, software coding and computer networks. The efficiency of using data structures in performing various operating system jobs are explored in detail with examples.

Index Terms—Data structures, operating system, applications, computer science, linear, non-linear.

1. INTRODUCTION

The heart of a system is its data, which is entered through a system's input, and is subsequently stored in files [1]. Data is simply assumed as values or set of values [2], on the other hand in database terminology data is also defined as collection of raw facts. Data individually do not carry any meaning but when logically related establishes meaningful information. The term "logically related" is more essential to gain better understanding of identification of data for storage and establishing logical relationships between them. A powerful and efficient storage mechanism is required for representation of data physically on the hard disk or computer's memory to preserve the logical relationships. A storage and organization structure is required which enables such efficiency by allowing the users (through necessary software) to manage all the data access or retrievals. Especially the way the data organized in memory is very essential to fulfill the needs of users through application software or system software like operating system.

Data may be organized in many different ways. The logical or mathematical model of a particular organization of data is called a "Data structure". An efficient data model is required to organize the data. The choice of particular data model depends on two considerations. Firstly it must be rich enough in structure to mirror actual relationships of data in real world and on the other hand, the structure should be simple enough that one can effectively process data when necessary.

This paper focuses on the applicational aspects of various data structures with relevance to the field of computer science and

also to system software like operating system and their ordeal in performing its operational tasks. The data organized with data structures are processed by means of certain operations. The decisive action of choosing a particular data structure for a given situation depends largely on frequency with which specific operations are performed. The most commonly performed operations on data structures include traversing, searching, inserting, deleting and few special operations like merging and sorting. The above operation performed without the employment of data structures in their algorithms leads to increased time complexity and also introduces many errors/bugs in the process and leaves out the process with many problems that remain unsolved. This shows an adverse effect on the performance of the system. The solution to all these problems is building the efficient algorithms adopting data structures in their algorithms appropriately for the situation leads to efficient management and maintenance of data by minimizing the complexity and making efficient use of system resources.

In organizing a solution to any problem solved with the aid of computer machine confronts with these four sub problems [3].

1. To understand thoroughly the relationships between data elements/items that are relevant to solution of problem.
2. To decide on the operations that must be performed on the logically related data elements.
3. To devise methods of representing the data elements in the memory of the computer such that the logical relationships which do exist between data items can best be retained and the operations on data elements/items can be accomplished easily and efficiently.
4. To decide on what problem solving language can best aid in the solution of the problem by allowing the user to express in natural manner the operations he/she wishes to perform on the data.

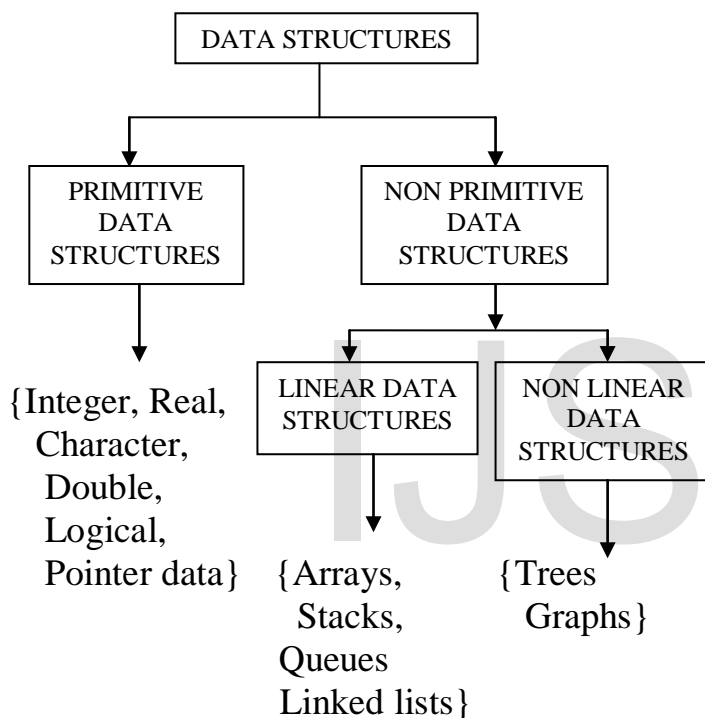
The representation of particular data structure in the system's memory is called a storage structure. Each and every data structure has different and multiple storage representations. Every data structure has a part of contribution in accomplishing functions of operating system and managing its tasks and resource utilization efficiently. For example, B-trees are particularly well-suited for implementation of databases [4, 2], while compiler implementations usually use hash tables to look up identifiers etc.

- Shobha Rani is currently pursuing Ph.D in Computer Science and technology in University of Mysore and working as lecturer in Computers in Amrita Vishwa Vidyapeetham, Mysore, India. E-mail: n.shoba1985@gmail.com
- Suman S.P is working as a lecturer in Computers in Amrita Vishwa Vidyapeetham, Mysore, India. E-mail: sumansppujar@gmail.com

A number of applications will be discussed in a comprehensive manner in the following section with necessary examples if required.

2. Various Data structures and its applications to Operating system:

Data structures are used in operational tasks of almost every program or software system. Some programming languages emphasize data structures rather than algorithms as the key organizing factor in software design. To begin with the overview of data structures and its applications, we find it necessary to classify clearly its various categories. The standard categorization of data structures include Primitive and non-primitive data structures [5]. The following figure shows the categories of data structures and its sub divisions.



A data structure is a specialized format for organizing and storing data as we mentioned in the above section. General data structure types include the array, the file, the record, the table, the tree, and so on. Any data structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways. In computer programming, a data structure may be selected or designed to store data for the purpose of working on it with various algorithms. Primitive data structures are the one that are directly manipulated by machine instructions. The primitive data structures include integers, real, logical data, character data and pointer data. The primitive data structures have its part of contribution to operating system in defining storage structures for different types of data. The storage structure of data types varies comparatively from one data type to another and also from operating software platform to another [6].

Non-primitive data structures are the one which are not directly manipulated by machine level instructions but are manually deployed into usage of any application requirements by means of building algorithms. Non-primitive data structures are again categorized into linear data structures and non-linear data structures. The relationship between linear and non-linear data structures is defined in terms of factor called principle of alignment. The principle of alignment represents whether data is stored adjacently or non-adjacently. Some of the linear data structures include arrays, structures, unions, stacks, queues and linked lists, files etc.

Linear data structures can be constructed as a continuous arrangement of data elements in the memory. It can be constructed by using array data type. In the linear Data Structures the relationship of adjacency is maintained between the Data elements.

The non-linear data structures are mainly trees and graphs. Non-linear data structure can be constructed as a collection of randomly distributed set of data item joined together by using a special pointer. In non-linear Data structure the relationship of adjacency is not maintained between the Data items.

The more advanced data structures may include dictionaries, skip lists and hash tables, heaps, leftist trees, winner trees, loser trees, balanced search trees like AVL trees, Red-black trees, splay trees and B-trees etc.

There are various important applications of multiple data structures associated with field of computer science which are as discussed below.

2.1 Arrays:

Arrays are homogenous and linear data structures that systematically arrange the data objects sequentially one after another in a continuous chunk of memory [7]. An array can hold several pieces of information of same type. Arrays are considered as building blocks of data structures. Arrays can be used to implement all the basic and advanced data structures that simplify the coding in many cases. Sometimes, arrays can even accomplish tasks which are not easily done with other methods. Array, besides its iterative and conditional processing, can be very powerful and have many applications.

Applications:

Array is like a universal data structure which have its relevance towards building all other data structures.

- ➔ To search and find a specified target value, such as maximum, minimum, mean, median, average, count from a group of data objects the arrays are very efficient.

- Array data structure is used for arrangement of items at equally spaced and sequential addresses in computer memory makes it easier to perform operations like sorting, merging, traversal, retrievals [8].
- Array data type, used in a programming language to specify a variable that can be indexed.
- Numerical representations like matrices can be easily represented in computer's memory which enables to solve many complex mathematical problems and images processing transformations.

2.2 Stacks:

Stack is homogenous linear and recursive data structure which accepts values in last in first out order[9]. Therefore they are called LIFO lists. The operation of inserting an element into the stack represents the PUSH and deleting an element from the stack represents the POP operation. A stack is a limited access data structure where elements can be added and removed from the stack only at the top. A helpful analogy is to think of a stack of books; you can remove only the top book, also you can add a new book on the top. The simplest application of a stack is to reverse a word. You push a given word to stack - letter by letter - and then pop letters from the stack.

Some of the applications of stacks associating in accomplishing operating system tasks include.,

- A stack is useful for the compiler/operating system to store local variables used inside a function block, so that they can be discarded once the control comes out of the function block.
- A stack can be used as an "undo" mechanism in text editors; this operation has been accomplished by keeping all text changes in a stack.
- Stacks are useful in backtracking, which is a process when you need to access the most recent data element in a series of elements.
- An important application of stacks is in parsing. For example, a compiler must parse arithmetic expressions written using infix notation:
$$1 + ((2 + 3) * 4 + 5) * 6$$
- It can be used to process function calls and implementing recursive functions. The return values and addresses of the function will be pushed into the stack and the lastly invoked function will first return the value by popping the stack.
- In language processing, space for parameters and local variables is created internally using a stack. Compiler's syntax check for matching braces is implemented by using stack.

2.3 Queues:

Queue is a linear and homogenous data structure in which insertions and deletions are done at different ends in First in first out order (FIFO). Hence they are called FIFO lists[10]. In queue all the elements are inserted at rear end where as all deletions are performed at another end called as front end. General applications of queues include transport, and operations research where various entities such as data, objects, persons, or events are stored and held to be processed later. In these contexts, the queue performs the function of a buffer.

Some important applications of queues in performing operational tasks of system include.,

- Queues are common in programs, where they are implemented as data structures coupled with access routines, as an abstract data structure or in object-oriented languages as classes. Common implementations are circular buffers and linked lists.
- Queues can be used to store the interrupts in the operating system.
- It is used by an application program to store the incoming data.
- Queue is used to process synchronization in Operating System
- Queues are used for CPU job scheduling and in disk scheduling.

This data structure is usually used in simulation of various features of operating system like multiprogramming platform systems and in different type of scheduling algorithms are implemented using queues. Printer server routines, various applications software are also based on queue data structure.

Example: Round robin technique is implemented using queues. . It is used especially for the time-sharing system. The circular queue is used to implement such algorithms.

2.4 Linked lists:

A linked list is a linear data structure consisting of a group of nodes, each node is composed of a data and a *link* to the next node in the sequence [11]. A linked list organizes stored data on a storage medium based on the logical order of the data and not the physical order.

Linked list is most useful linear data structure it has various applications in performing many operational tasks with system.

- Linked lists are used in dynamic Memory Management tasks like allocation and releasing memory at run time.

- A polynomial can be represented and manipulated using linear link list to perform operations such as addition, multiplication with polynomials.
- Linked lists are used in Symbol Tables for balancing parenthesis and in representing Sparse Matrix.
- A linked list would be a reasonably good choice for implementing a linked list of file names, undo functionality in Photoshop.
- The cache in your browser that allows you to hit the BACK button where a linked list of URLs can be implemented.
- Data structures like stack, hash table, and binary tree can be implemented using a doubly linked list.
- A linked list is simply the way to manage unbounded memory in system.

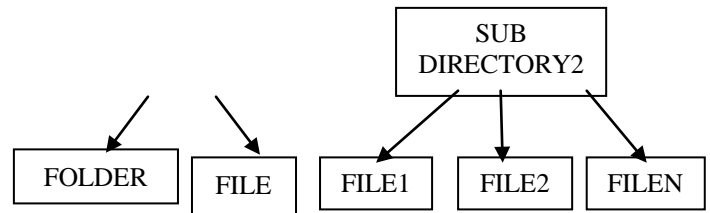
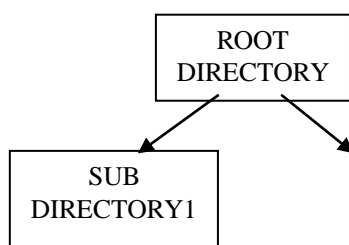
Some common applications of linked lists include creating hash tables for collision resolution across communication channels, structuring binary trees, building stacks and queues in programming, and managing relational databases.

2.5 Trees:

A *tree* data structure is a powerful tool for organizing data objects based on keys. It is equally useful for organizing multiple data objects in terms of hierarchical relationships. Tree structures make an excellent alternative to arrays, especially when the data stored within them is keyed or has internal structure that allows one element to be related to, or "saved within" another.

Applications:

- Trees are used to represent phrase structure of sentences, which is crucial to language processing programs. Java compiler checks the grammatical structures of Java program by reading the program's words and attempting to build the program's parse tree. If successfully constructed the parse tree is used as a guide to help the Java compiler in order to generate the byte code that one finds in program's class file.
- Trees are used in many search applications where data is constantly entered and deleted such as the maps and set objects of many language libraries.
- An operating system maintains a disk's file system as a tree, where file folders act as tree nodes. The tree structure is useful because it easily accommodates the creation and deletion of folders and files.



Similarly, a menu system also has the same idea of selecting progressive layers of options.

- Tree data structures can be used for syntax validation in major compilers and as implementations of sorted dictionary.
- Trees have vast applications in computer networking and are used commonly in internet protocols. Trees can be used as every high-bandwidth router for storing router-tables.
- Trees can be searched using Dijkstra's algorithm to find nodes that are closest to the router for the shortest paths.
- Trees can be used for quick traversals and searching of directory structures in the system.

Trees are varied in its types and varieties, all together has multiple applications in different fields.

2.6 Graphs:

A graph is a data structure of a finite set of ordered pairs, called edges or arcs, of certain entities called nodes or vertices.

A graph may be undirected or directed from one vertex to another. Many practical problems in computer science can be represented and solved by graphs. Graph theoretical ideas are highly utilized in the field of computer science [12]. Modeling of data structures with vertices and edges to solve problems like resource allocation, scheduling, graph coloring, resource networking, database design, network topologies etc are considerably simpler and easier with graphs.

Some of the practical applications of graphs in the field of computer science include.,

Applications:

- The link structure of a website could be represented by a directed graph: the vertices are the web pages available at the website and a directed edge from page A to page B exists if and only if A contains a link to B.
- Graph coloring concept can be applied in job scheduling problems of CPU, jobs are assumed as vertices of the graph and there will be an edge between two jobs that cannot be executed simultaneously and there will be one-one relationship between feasible scheduling of graphs.

- Similarly, simultaneous execution of jobs problem between set of processors and set of jobs can be easily solved with graphs.

Example: A processor cannot work on two jobs simultaneously. This type of tasks will arise when scheduling of file transfers occurs between processors. This can be modeled by considering a graph whose vertices correspond to the processes and if there is any task that has to be executed on processors i and j , then an edge to be added between the two vertices. Now the scheduling problem is to assign colors to edges in such a way that every color appears at most once at a vertex.

- The problem of complex time table scheduling can be dealt efficiently with graphs. A bipartite graph G where the vertices are the number of instructors say $m_1, m_2, m_3, m_4, \dots, m_k$ and n number of subjects say $n_1, n_2, n_3, n_4, \dots, n_m$ such that the vertices are connected by p_i edges. It is presumed that at any one period each professor can teach at most one subject and that each subject can be taught by maximum one professor.

Example: Consider a period setting the timetable for this single period corresponds to a matching of possible assignment of professors to subjects taught during that period. Hence, the solution for the time table scheduling problem will be obtained by partitioning the edges of graph G into minimum number of matching. This problem can also be solved by vertex coloring algorithm.

- Shortest Path algorithm is the commonly used algorithm for determining the shortest path between any two points. This is practically used in applications such as network stations, computer games, maps, satellite navigation system etc.
- Minimal Spanning Tree algorithm is used to estimate the minimal cost or cheapest way to connect any two or more point such as networks, distance between two locations in memory, distance between as two cities, telephone or electrical stations etc.
- Optimal Graph Coloring algorithm is used to color a map with a minimum number of colors. A vertex coloring is a labeling of the vertices of an undirected graph such that no edge has two endpoints with the same label. The coloring algorithm visits every vertex, identifying the minimal number of colors necessary to label every vertex.

Example: Scheduling and resource allocation using a resource conflict graph. Vertices in a resource conflict graph represent operations to perform; edges represent pairs of operations which are in conflict because they cannot use the same resource.

- Graphs also have its applications in searching an element in a data structure using DFS or BFS.

Graphs have still many more practical applications applicable to other fields and which have not discussed in the scope of this paper. Graphs have numerous applications in solving network problems.

2.7 Some real time applications of data structures:

Some real time applications of data structures are

- Determination of cities using Google maps to find population.
- Elevation and location of the data structures supports in performing operations like finding addresses on the map associated with some conditions like lookup city by name, calculates shortest path between two cities.
- To display cities within a given window, insert, delete or rename cities etc.

Applications of data structures for Databases:

Complex data structures are essential in the design of fast algorithms for a variety of applications, including combinatorial optimization, information retrieval and Web search, databases and data mining, and geometric applications. The data structures employed in a DBMS context are B-trees, buffer trees, quad trees, R-trees, interval trees, hashing etc. Data Structures for Query Processing high-level input query expressed in a declarative language called SQL [2] the parser scans, parses, and validates the query.

2.8 Applications of data structures in Software Development:

The data structures and algorithms are concerned with the coding phase of software development. The use of data structures and algorithms is the nuts-and-bolts used by programmers to store and manipulate data.

3. Conclusion:

This paper is designed to benefit the students of computer science to gain depth knowledge on data structures and its relevance with other subjects like operating systems, Networks, Databases, software engineering etc. We have considered the various applications of major data structures that have relevance to the field of computer science and applications. The theme of this paper is to highlight all the vital applications of data structures related to operating system's functions. Data structures are considered as building blocks of efficient algorithms to deal with

operating system functions. The importance of each data structure in performing functions like resource allocation, job or process scheduling, organization of data in memory, context switching between processes, establishing relationship between processes or entities etc are reviewed in detail. The association between operating system and data structures, data structures and field of computer science has been explored well in this paper with necessary examples.

REFERENCES

- [1] Mohammad A., "Understanding data structures in system analysis and design course", University of Houston-Clear Lake.
- [2] Seymour Lipschutz. " Theory and problems of data structures", Tata Mc graw hill international editions".
- [3] Jean-paul Tremblay, Paul G. Sorenson. " an introduction to data structures with application", Second edition, Tata Mc. Graw hill.
- [4] A.M. Padma Reddy. "Systematic approach to data structures using C".
- [5] Sartaj sahni, "Data structures, algorithms and applications in C++", Second edition, University press.
- [6] http://en.wikipedia.org/wiki/Data_structure
- [7] [9] Zaizai., "Advanced Array Applications in Clinical Data Manipulation", AstraZeneca Pharmaceuticals David Shen, WCI, Inc.
- [8] <http://en.wikipedia.org/wiki/Array>.
- [9] <http://php.about.com/od/learnphp/a/Understanding-Arrays-In-Php.htm>.
- [10] Steve First, Teresa Schudrowitz., "Arrays Made Easy: An Introduction to Arrays and Array Processing", Systems Seminar Consultants, Inc., Madison, WI.
- [11]<http://people.cis.ksu.edu/~schmidt/300s05/Lectures/Week7b.html>.
- [12] S.G.Shirinivas, S.Vetrivel, Dr. N.M.Elango, " The applications of graph theory in computer science an – overview", International Journal of Engineering Science and Technology, Vol. 2(9), 2010, 4610-462.