Chief Editor
Dr. A. Singaraj, M.A., M.Phil., Ph.D.
Editor
Mrs.M.Josephin Immaculate Ruba

EDITORIAL ADVISORS

1. Prof. Dr.Said I.Shalaby, MD,Ph.D.
   Professor & Vice President
   Tropical Medicine,
   Hepatology & Gastroenterology, NRC,
   Academy of Scientific Research and Technology,
   Cairo, Egypt.
2. Dr. Mussie T. Tessema,
   Associate Professor,
   Department of Business Administration,
   Winona State University, MN,
   United States of America,
3. Dr. Mengsteab Tesfayohannes,
   Associate Professor,
   Department of Management,
   Sigmund Weis School of Business,
   Susquehanna University,
   Selinsgrove, PENN,
   United States of America,
4. Dr. Ahmed Sebihi
   Associate Professor
   Islamic Culture and Social Sciences (ICSS),
   Department of General Education (DGE),
   Gulf Medical University (GMU),
   UAE.
5. Dr. Anne Maduka,
   Assistant Professor,
   Department of Economics,
   Anambra State University,
   Igbariam Campus,
   Nigeria.
6. Dr. D.K. Awasthi, M.SC., Ph.D.
   Associate Professor
   Department of Chemistry,
   Sri J.N.P.G. College,
   Charbagh, Lucknow,
   Uttar Pradesh. India
7. Dr. Tirtharaj Bhoi, M.A, Ph.D,
   Assistant Professor,
   School of Social Science,
   University of Jammu,
   Jammu, Jammu & Kashmir, India.
8. Dr. Pradeep Kumar Choudhury,
   Assistant Professor,
   Institute for Studies in Industrial Development,
   An ICSSR Research Institute,
   New Delhi- 110070, India.
9. Dr. Gyanendra Awasthi, M.Sc., Ph.D., NET
   Associate Professor & HOD
   Department of Biochemistry,
   Dolphin (PG) Institute of Biomedical & Natural Sciences,
   Dehradun, Uttarakhand, India.
10. Dr. C. Satapathy,
    Director,
    Amity Humanity Foundation,
    Amity Business School, Bhubaneswar,
    Orissa, India.

ISSN (Online): 2455-7838
SJIF Impact Factor (2017): 5.705

EPRA International Journal of Research & Development (IJRD)

Monthly Peer Reviewed & Indexed
International Online Journal

Volume: 3, Issue:9, September 2018

Published By :
EPRA Journals

CC License
BY NC ND

SJIF Impact Factor: 5.705 Volume: 3 | Issue: 9 |September| 2018 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)

AN APPLICATION OF DATA STRUCTURES IN ARTIFICIAL INTELLIGENCE, ABSTRACT DATA TYPE

R.S.Vipan Kumari[1]
Himalayan University, Itanagar, Arunachal Pradesh, India

Sandeep Kulkarni[2]
Himalayan University, Itanagar, Arunachal Pradesh, India

ABSTRACT
Data structure and artificial intelligence that is used constantly in many search problems in Artificial Intelligence is the priority queue. This data structure is implemented usually with heaps. This data structure is particularly useful in the search algorithm, which is used to find shortest paths in graphs using admissible heuristics. The priority queue assigns each node with a weight and is used to determine the next node to check in the algorithm. It is the scientific and practical approach to computation and its applications and the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to, information. A pre-defined data type such as Integer, Strings, Boolean is used to create the structure of data. The use of data type is based on the user requirement and the kind of data they want to store. Now let's dive into the data structures that are used in our day to day programming, and we will also look into the support for the data structures for different programming languages.

KEYWORDS: *Artificial Intelligence, Data Structure, Algorithm, Structure Expression, Data Representation.*

1.0 INTRODUCTION

Artificial Intelligence and data structure is the study of the theory, experimentation, and engineering that form the basis for the design and use of computers. It is the scientific and practical approach to computation and its applications and the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to, information. An alternate, more succinct definition of computer science is the study of automating algorithmic processes that scale. A computer scientist specializes in the theory of computation and the design of computational systems. See glossary of computer science. Its fields can be divided into a variety of theoretical and practical disciplines. Some fields, such as computational complexity theory(which explores the fundamental properties of computational and intractable problems), are highly abstract, while fields such as computer graphics emphasize real-world visual applications. Other fields still focus on challenges in implementing computation. Describing data structures as a way of organizing and storing data so that operations can be performed efficiently. Accessing, inserting, deleting, finding, and sorting the data are some of the basic operations that one can perform using data structures. Not all data structures can perform these operations efficiently, that's what led to the development of different data structures.

1.1 Data Structure

DATA STRUCTURE is a collection of data elements whose organization is characterized by assessing operations that are used to store and retrieve the individual data elements; the implementation of the composite data members in an abstract data type. An abstract data type can be defined as a data type whose properties are specified independently of any particular implementation. Data structures can implement one or more particular abstract data types (ADT), which specify the operations that can be performed on a data structure and the computational complexity of those operations. In comparison, a data structure is a concrete implementation of the space provided by an ADT. Different kinds of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks. For example, relational databases commonly use B-tree indexes for data retrieval, while compiler implementations usually use hash tables to look up identifiers. Data structures provide a means to manage large amounts of data efficiently for uses such as large databases and internet indexing services. Usually, efficient data structures are key to designing efficient algorithms. Some formal design methods and programming languages emphasize data structures, rather than algorithms, as the key organizing factor in software design. Data structures can be used to organize the storage and retrieval of information stored in both main memory and secondary memory.

1.1.1 Abstract Data Type:-

In computer science, an **Abstract Data Type** (**ADT**) is a mathematical model for data types, where a data type is defined by its behavior (semantics) from the point of view of a use of the data, specifically in terms of possible values, possible operations on data of this type, and the behavior of these operations. This contrasts with data structures, which are concrete representations of data, and are the point of view of an implementer, not a user.

An abstract data type can be defined as a data type whose properties are specified independently of any particular implementation. The DATA STRUCTURE is classified in the following categories:

1.1.2 Linear Data Structures:- In the linear Data Structures processing of data items is possible in linear fashion, i.e., data can be processed one by one sequentially. Linear data structures contain following types of data structures.

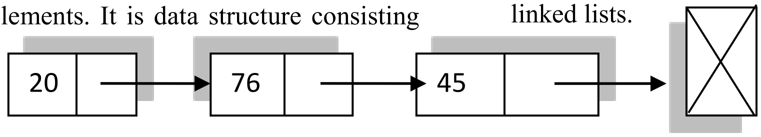
a) Array:- An array is a variable which can store multiple values of same data type at a time. The collection of the similar data items stored in continuous memory location with single name. Whenever we want to work with large number of data values, we needed to use that much number of different variables. The number of variables are increasing, complexity of the program are also increases.

As you know, these particular data types beneficial for declaring variables, constant or a return type for a function; they are in control by the fact that, these types can be store only a specific form of a value at a time. For many applications, there types can store only a specific from of value at a time. The syntax for declaring array is:-

```
Data_type array_ name [array_size] ;
```

The programmers must need to have the fundamental data type's name such as char, int, float & double.

b) Linked list:- A Linked list is a linear collection of the data elements. It is data structure consisting of a collection of nodes which together represent a sequence. Each node contains : data and a references to the next node in the sequence. A drawback of linked list in that access time is linear. The array have better cache locality compared to the linked lists.



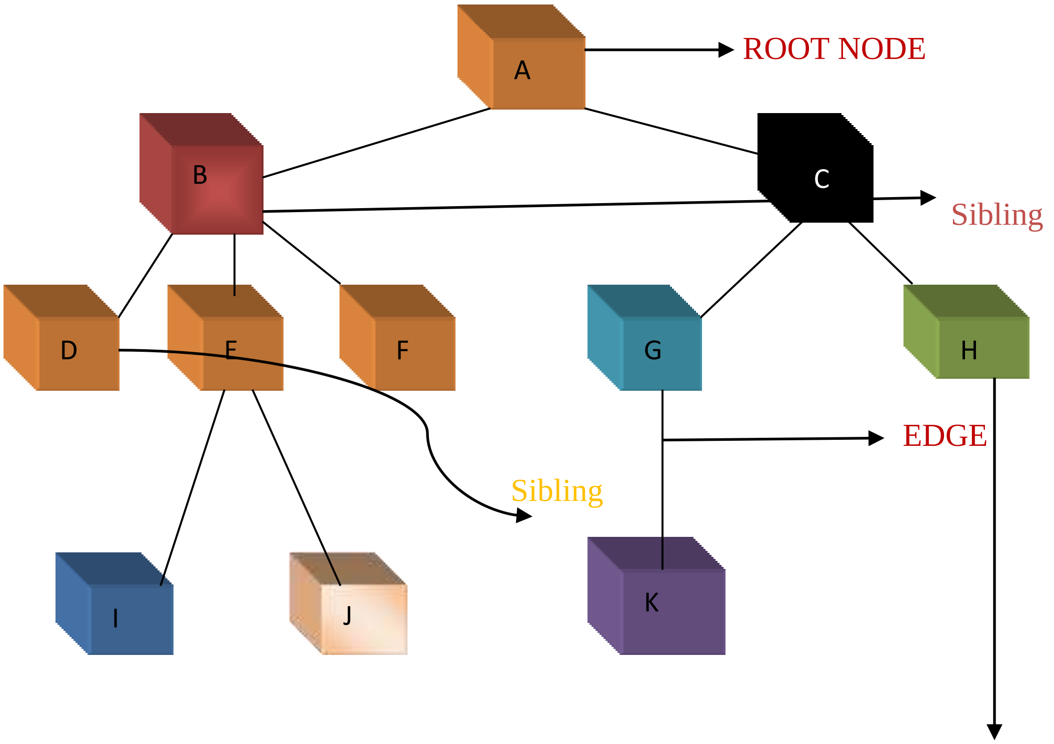
The linked list whose nodes contain two fields:- An integer value and link to the next node. The last node is linked to a terminator used to signify the end of the list. The linked list are among the simplest and most common data structures. They can be used to implement several other common abstract data types, including list , stacks, queues etc.

c) Stack :- Uses Artificial intelligence a stack is an Abstract Data Type (ADT), It commonly used in most programming languages. It is name stack as it behaves like a real-world stack, e.g:- a collection of balls or piles of plates etc. A real world stack allows operations at one and only. Such as , we can place or remove a ball or plate from the top of the stack only.

At any given time, we can only access the top element of a stack. This features makes it LIFO data structure. LIFO stands for Last-in-first-out. The element which is insert or added last, is accessed first. In stack terminology, insertion operation is called PUSH operation and removal operation is called POP operation.

1.1.3 **Nonlinear Data Structures**:- A Data Structures in which insertion and deletion is not possible in a linear fashion is called nonlinear DATA STRUCTURE. In this category of DATA STRUCTURE, we will discuss the following.

a) **Tree: -** A tree is a non- linear data structure which organizes data in hierarchical structure. The tree data structure is a collection of data (node) which is organized node is a recursive definition. A tree data structure, every individual element is called node. Node in a tree data structure, stores the actual data of that particular element and link to next element in hierarchical structure. In a tree data structure, if we have N number of nodes then we can have a maximum of N-1 number of links. Example:-



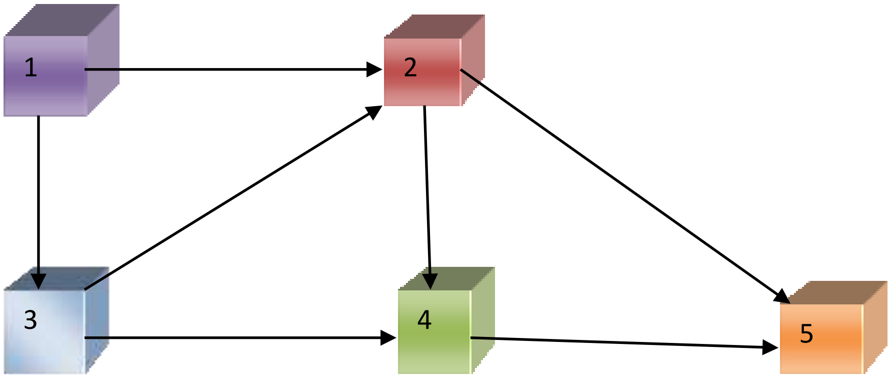
Descendant of any node is

Called CHILD Node

Here, ‘A’ is Root Node,

A, B, C, E, G is Parent Node,
Here, B and C are children of A, G and H are children of C, K is child of G
Here, D, I, J, K, & H are Leaf Nodes, A,B,C,E & G are Internal Node.
This tree with 11 nodes and 10 edges. In any tree with ‘N’ nodes there will be maximum of ‘N-1’ edges. Ina tree every individual element is called ‘NODE’

b) Graph:- In artificial intelligence using Data Structure with graph is a pictorial representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are represented as Vertices, Edges.
Therefore, a graph is (V, E), where V is the set of vertices of E is the set of edges, connecting the pair of vertices. Following graph are:-



In above graph,
V={1,2,3,4,5}
E={2.4, 1.2,1.3,3.4,4.5,3.2} A graph data structure consist of a finite set of vertices or nodes or points, together with a set of unordered pairs of these vertices for an undirected graph or a set of o0rdered pairs for a directed graph. These pairs are known as edges.

1.1.4 Types of Data Structure by Using Artificial Intelligence:-

1. **Primitive Data Structure**:-Primitive Data structure: This is the data structures that typically are directly operated upon by machine-level instructions. We will present storage representations for these data structures for a variety of machines. For example, declarations of variables with basic data types are the example of primitive data structure.
2. **Non-Primitive Data structure:** Non-primitive data structures can be classified as arrays, lists and files. As array is an ordered set which consists of a fixed number of objects. No deletion or insertion operations are performed on arrays. A list, on the other hand, is an ordered set consisting of a variable number of elements to which insertions and deletions can be made.

2.1 IMPLEMENTATION OF DATA STRUCTURE USING AI

Data structures are generally based on the ability of a computer to fetch and store data at any place in its memory, specified by a pointer—a bit string, representing a memory address, that can be itself stored in memory and manipulated by the program. Thus, the array and record data structures are based on computing the addresses of data items with arithmetic operations; while the linked data structures are based on storing addresses of data items within the structure itself. Many data structures use both principles, sometimes combined in non-trivial ways (as in XOR linking).

The implementation of a data structure usually Artificial Intelligence by requires writing a set of procedures that create and manipulate instances of that structure. The efficiency of a data structure cannot be analyzed separately from those operations. This observation motivates the theoretical concept of an abstract data type, a data structure that is defined indirectly by the operations that may be performed on it, and the mathematical properties of those operations (including their space and time cost).

2.1.1 Types of Artificial Intelligence Systems using Data Structure:-

1. Machine Learning

Early researchers struggled with limited processing power and computer storage, but still laid the foundation of AI with programming languages like LISP and concepts like decision trees and machine learning. Programs written in LISP could easily analyze games like chess, map all possible moves for several turns, then choose the best alternative. These programs could also modify their decision logic and learn from previous mistakes, getting "smarter" over time. With more powerful computers and cheaper mass storage, this branch of AI spawned the computer gaming industry, as well as a variety of personalized search engines and online shopping sites that not only remember our preferences, but anticipate our needs.

2. Expert Systems

While the first wave of AI researchers relied on computing cycles to simulate human reasoning, the next approach relied on facts and data to mimic human experience. Expert systems gathered facts and rules into a knowledge base then used computer-based inference engines to deduce new facts or answer questions. Knowledge engineers interviewed experts in medicine, automotive repair, industrial design or other professions, then reduced these findings into machine readable facts and rules. These knowledge bases were then used by others to help diagnose problems or answer questions. As the technology matured, researchers found ways to automate knowledge base development, feeding in reams of technical literature, or letting the software crawl the Web to find relevant information on its own.

3. Neural Networks

Another group of researchers tried to reproduce the workings of the human brain by creating artificial networks of neurons and synapses. With training, these neural networks could recognize patterns from what looked like random data. Images or sounds are fed into the input side of the network, with the correct answers fed into the output side. Over time, the networks reorganize their internal structure so that when a similar input gets fed in, the network returns the correct answer. Neural networks work well when responding to human speech or when translating scanned images into text. Software that relies on this technology can read books to blind people or translate speech from one language to another.

4. Big Data

Large scale data analysis, often called "big data," harnesses the power of many computers to discover facts and relations in data that the human mind cannot comprehend. Trillions of credit card charges or billions of social network relations can be scanned and correlated using a variety of statistical methods to discover useful information. Credit card companies can find buying patterns that indicate that a card has been stolen, or that a cardholder is in financial difficulty. Retail merchants may find buying patterns that indicate that a customer is pregnant, even before she knows this herself. Big data allows computers to understand the world in ways that we humans never could on our own.

3.1 CONCLUSION

Data Structures is not just limited to Stack, Queues, and Linked Lists but is quite a vast area. There are many more data structures which include Maps, Hash Tables, Graphs, Trees, etc. Each data structure has its own advantages and disadvantages and must be used according to the needs of the application. A computer science student at least knows the basic data structures along with the operations associated with them.

Many high level and object oriented programming languages like C#, Java, Python come built in with many of these data structures. Therefore, it is important to know how things work under the hood.

REFERENCES

1. *[2018] edition of by Wiki Loves "Data Structure used by Artificial Intelligence" and development.*
2. *[2017] A computer Science Portal of Greeks for Greeks "overview of data structure, stacks, overview of linked list & Abstract data type".*