ISSN No. 0976-5697

Volume 6, No. 6, July-August 2015

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# Solve Some of the Problems of Linked List

Reza Jafari Ziarani
Department Of Computer Science
WTIAU College
Tehran, Iran

Payam Morad Abadi
Department Of Computer Science
WTIAU College
Tehran, Iran

Mohammad Amin Madadi
Department Of Computer Science
WTIAU College
Tehran, Iran

***Abstract:*** Current data structures have some problems that the purpose of this paper is to resolve some of these problems. One of these problems is the include lack of random access or the need for additional space for pointers in the linked list.
To solve these problems we propose the GDS (Graphical Data Structures) matrix. This matrix is actually another way to implement the linked list. One of the features of GDS matrix is that it can store any data the way it is. For example we implement the binary tree using the linked list but in the GDS matrix we implement the tree as it is drown on the paper (This example is illustrated in the results)

***Keywords:*** Data structures, Linked list problems, Binary tree, Matrix, GDS Matrix

## I. INTRODAUCTION

Due to the increasing need for information systems, the lack of a flexible and simple data structure is felt.

Currently we have to form the real data in a way in which we can express them with existing data structures. This is considered as a limit to the progress of science because first it is necessary to create its data structure or change the data in a way that they could be expressed by the current data structures.

In general there are two main problems in current data structure including:

1. To change existing data structures and create a new data structure a professional and experienced expert in the field of computer is required.
2. There is a huge gap between the real and virtual entities.

For example, consider the image of a chair and its binary model. Obviously by looking at the equivalent binary it is not possible to understand the code belongs to which object. This is the gap between the real and virtual entities.

There are also specific problems related to each structure. For example the following problems are in the linked list. [1]

1. Waste of memory
2. The lack of random access
3. The difficulty of reversing
4. The time required for implementation and reading
5. Heap space restrictions

In order to implement tree data structures the linked list is used as well and that is why the problems are shared.

An approach using hashing is presented for linked lists. [2] This method is designed based on the fact that parts of the tree that are filled with sub-trees can be presented by the hash table. This table includes the leaves of the sub-trees. [3] This is performed without the need for memory and no data loss occurs.

This simple idea reduces the number of memory reference to log n. n is the size of the tree. [4]

Our main goal is to express a data structure to solve linked list problems. The importance of the problem is that most phenomena around us suffer from disorder (entropy) and that is why they are expressed by linked lists.

## II. GDS MATRIX

GDS matrix is actually a two-dimensional array which is used to implement the linked list or in artificial intelligence. In this method the definition of pointers is prevented by defining the matrix reading. This will reduce the volume and high speed to read the data.

For storing GDS matrix P2C or P4C formats are used frequently. In short P2C method uses a bit to express each pixel and P4C method uses two bits to express each pixel. These formats are already discussed and approved at the national conference. [5]

To better understand this matrix, consider the following example:

The following binary tree is supposed. We want to implement and compare this binary tree once with liking list and once with GDS (Graphical Data Structures).

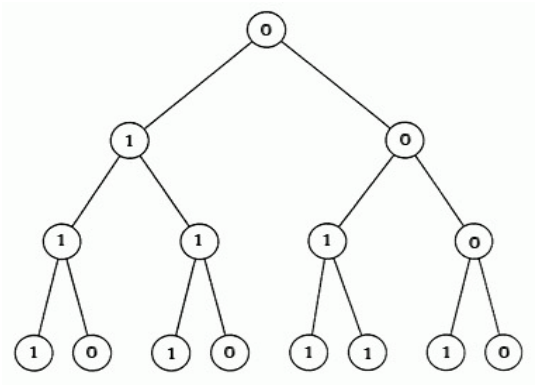


Figure 1. the tree structure

Table I. GDS data structure

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 00 | | | | | |
| | | | | 11 | | 00 | | | | |
| | | 11 | 11 | | | | 11 | 00 | | |
| 11 | 00 | 11 | 00 | | | | 11 | 11 | 11 | 00 |

We fill the empty places with code 01 which is gray, 00 is for white and 11 is for black

Table II. The guide to convert binary code to the GDS code

| *Colors in graphical data structure* | *GDS data structure code* | *Binary code* |
|---|---|---|
| White | 00 | 0 |
| Black | 11 | 1 |

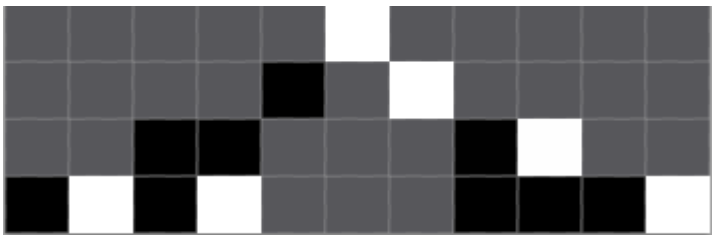

Figure 2. Final graphical data structure

Figure 2 is the Final graphical data structure that is stored and the required process is done on it.

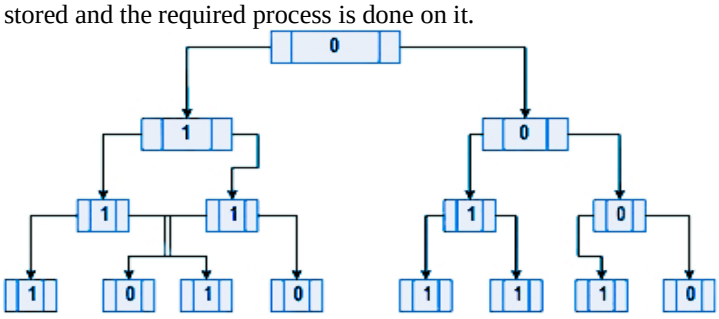


Figure 3. Displaying the tree using a linked list

It might seem at first glance that the image takes more space than the linked list, but that is not true. A pointer is used in the linked list. The pointers include the next elements and they must occupy at least one byte space. If we consider the space required for each address one byte the following calculations indicate that GDS occupies less space.

The required space for GDS:

44 (number of pixels) ×2 bits (space for each pixel) = 88 bits = 11 bytes

The required space for linked list:

14 bytes (pointer) + 15 bytes (the data) = 29 bytes

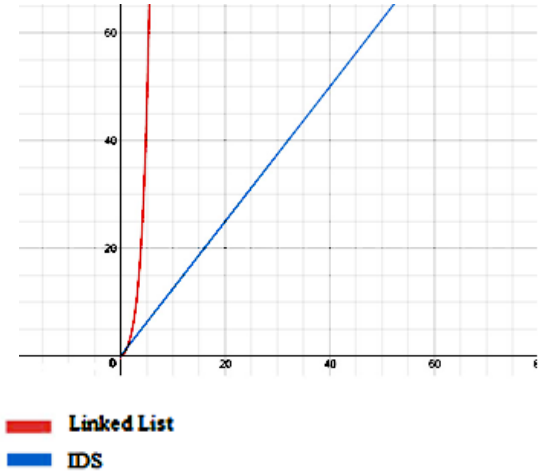


Figure 4. comparing the growth rate of linked list and GDS for binary tree

## III. Conclusion

According to figure (4) and the possibility of random access in the proposed method, the use of this method of implementation is more suitable.

## IV. References

[1] Juan,Angel,"Linked list", https://en.wikipedia.org/wiki/Linked_list#Disadvantages ,(February 2014).

[2] M. Waldvogel, G. Varghese, J. Turner, B. Plattner, "Scalable High-Speed IP Routing Lookups", Proceedings of SIGCOMM '97 (October).

[3] Ananth Grama, and Mikhail,"Adaptive Data Structures for IP Lookups Ioannis Ioannidis" , Purdue University, W. Lafayette, IN 47907 (2008).

[4] S. Nilsson, G. Karlsson, "Fast Address Look-Up for Internet Routers", Proceedings of IEEE Communications Magazine (January 2011).

[5] Reza Jafari Ziarani , "Compression algorithm for images",4th communication & Telecommunication Applied science & Technology Conference, 2015, Shiraz, I. R. Iran.