

Chapter 12

Heterogeneous Data Structure “r–Atrain”

Ranjit Biswas
Jamia Hamdard University, India

ABSTRACT

The data structure “r-Train” (“Train” in short) where r is a natural number is a new kind of powerful robust data structure that can store homogeneous data dynamically in a flexible way, in particular for large amounts of data. But a train cannot store heterogeneous data (by the term heterogeneous data, the authors mean data of various datatypes). In fact, the classical data structures (e.g., array, linked list, etc.) can store and handle homogeneous data only, not heterogeneous data. The advanced data structure “r-Atrain” (“Atrain” in short) is logically almost analogous to the data structure r-train (train) but with an advanced level of construction to accommodate heterogeneous data of large volumes. The data structure train can be viewed as a special case of the data structure atrain. It is important to note that none of these two new data structures is a competitor of the other. By default, any heterogeneous data structure can work as a homogeneous data structure too. However, for working with a huge volume of homogeneous data, train is more suitable than atrain. For working with heterogeneous data, atrain is suitable while train cannot be applicable. The natural number r is suitably predecided and fixed by the programmer depending upon the problem under consideration and also upon the organization/industry for which the problem is posed.

INTRODUCTION

Data structures are generally based on the ability of a computer to fetch and store data efficiently at any place in its memory. The data structures record and array 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. Data structures can be classified in several ways, viz. Primitive/Non-primitive, Homogeneous/Heterogeneous, Static/Dynamic,

DOI: 10.4018/978-1-4666-4936-1.ch012

Linear/Non-linear, etc. The cost of memory space to store data using the data structure array is least. At the same time a basic demerit of using array is that it needs contiguous free bytes of memory which may not be always available if the array size is large, even if huge amount of free bytes be available in the memory at different locations. For example, at some point of time it may happen that 2^{20} number of free bytes available in the memory whereas 2^7 free contiguous bytes are not available. Consequently, an array of 65 positive integers can not be accommodated in the memory at this point of time, whereas a linked list of a large number of integers can be accommodated easily here. This is one of the basic advantages of using linked list. However, linked lists require more memory space compared to arrays because of an extra field in each node where address of the next node is stored. There are a number of useful data structures viz. Splay Trees and Dynamic Trees (Sleator & Tarjan, 1983; Sleator & Tarjan, 1985), hash table, AVL Trees (Velskii & Landis, 1962), B-Trees (Bayer & McCreight, 1972), Red-black trees (Bayer, 1972), Treaps (Seidel & Aragon, 1996), HAT (Sitarski, 1996), Graphs, Multigraphs (Biswas, Alam, & Doja, 2012; Biswas, Alam, & Doja, 2013), etc. developed by different authors for various purposes with different philosophy for different kind of requirements (one could see any good book for details viz. Aho, Hopcroft, & Ullman, 1983; Shaffer, 2001; Knuth, 1968; Lynch, 1996; Cormen, Leiserson, Rivest, & Stein, 2001). *There are some situations where we need to create an entirely new type of data structure of generalized nature, and creating such a new but simple data structure is not always a straightforward task.*

In this chapter we discuss about two new powerful data structures “r-Train” (or “Train” in short) and “r-Atrain” (or “Atrain” in short) which are very suitable for storing huge volume of data. The natural number r is suitably pre-decided and fixed by the programmer depending upon the problem under consideration and also upon the organization/industry for which the problem is posed.

While train stores homogeneous data, atrain is suitable for the purpose of storing heterogeneous data. The term “Atrain” is an abbreviation for “Advanced train”. The two data structures r-train and r-atrain do not have any conflict of interest at all, although both are potential enough to accommodate huge volume of data. There is no competition between r-atrain and r-train in application areas, because r-train can not be applicable for heterogeneous data and r-atrain is not more suitable data structure than r-train if the data are homogeneous.

HOMOGENEOUS DATA STRUCTURE “r-TRAIN” (TRAIN)

The data structure ‘r-train’ (train, in short) is a new and a robust kind of dynamic homogeneous data structure which encapsulates the merits of the arrays and of the linked lists, and at the same time inherits a reduced amount of their characteristic demerits. By the nature ‘dynamic’ we mean that it changes over time, it is not like static. Apparently it may seem that the homogeneous data structure r-train is of hybrid nature, hybrid of linked list and array; but construction-wise (by virtue of its architecture) it is more. Using the data structure ‘r-train’, a large number of homogeneous data elements can be stored and basic operations like insertion/deletion/search can be executed very easily. In particular parallel programming can be done very easily in many situations with improved time complexities and performance efficiency. The notion of r-train is not a direct generalization of that of the linked lists. It is not just a linked list of arrays as it looks so, but something more. However, every linked list is an example of 1-train (i.e. r-train with $r = 1$). The main aim of introducing the data structure ‘r-train’ is how to store a large or very large array in memory in an alternative way successfully, and how to perform the fundamental operations of data structures efficiently while the data are huge in count. For small size arrays it is

20 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/heterogeneous-data-structure-r-atrain/97064

Related Content

Cognitive Memory: Human Like Memory

Bernard Carlos Widrow and Juan Aragon (2010). *International Journal of Software Science and Computational Intelligence* (pp. 1-15).

www.irma-international.org/article/cognitive-memory-human-like-memory/49128

TAntNet-4: A Threshold-Based AntNet Algorithm with Improved Scout Behavior

Ayman M. Ghazy and Hesham A. Hefny (2017). *Handbook of Research on Machine Learning Innovations and Trends* (pp. 942-974).

www.irma-international.org/chapter/tantnet-4/180980

Using Clustering for Forensics Analysis on Internet of Things

Dhai Eddine Salhi, Abekamel Tari and Mohand Tahar Kechadi (2021). *International Journal of Software Science and Computational Intelligence* (pp. 56-71).

www.irma-international.org/article/using-clustering-for-forensics-analysis-on-internet-of-things/266228

Multi-Fractal Analysis for Feature Extraction from DNA Sequences

Witold Kinsner and Hong Zhang (2012). *Breakthroughs in Software Science and Computational Intelligence* (pp. 100-118).

www.irma-international.org/chapter/multi-fractal-analysis-feature-extraction/64605

A Novel Chaotic Northern Bald Ibis Optimization Algorithm for Solving Different Cluster Problems [ICCICC18 #155]

Ravi Kumar Saidala and Nagaraju Devarakonda (2019). *International Journal of Software Science and Computational Intelligence* (pp. 1-25).

www.irma-international.org/article/a-novel-chaotic-northern-bald-ibis-optimization-algorithm-for-solving-different-cluster-problems-iccicc18-155/233520