DATA POOL: A STRUCTURE TO STORE VOLUMINOUS DATA

Dhanashri V. Sahasrabuddhe and Dr. Pallavi P. Jamsandekar
Bharati Vidyapeeth Deemed University, Institute of Management and Rural Development Administration, Sangli, Maharashtra, India.

ABSTRACT

It is era of ‘Data Flood’ which has given rise to Big Data. The paper presents a data structure named ‘Data Pool’ to represent big data in memory while carrying out process. In the paper researcher has discussed implementation and results of the implementation with respect to space and time complexity of the structure. The structure is implemented for weather data. For weather forecasting process weather data is given as input using file formats, which is arranged in specific format. Before passing data as input, the structure can be applied to represent the data, which enables to represent maximum possible weather data with minimum possible memory space.

To generalize the structure researchers implemented the structure for health data. The paper discusses results of the implementation with reference to space (memory) reduction as compared to original data, time consumed while processing data in Data Pool, and significant data loss with reference to original data and data represented in Data Pool.

Keywords Big Data, Data Structure, Weather Data, Health Data, Complexity of the structure


1. INTRODUCTION

The paper presents a new data structure which can be used to store data which is comparatively big in size. Before working on development of the structure, referred as Data Pool, researcher had undertaken review of literature for data structures and big data with reference to weather forecasting process. After reviewing various literature, to conclude with literature review, a need for data structure was realized which may help to improve weather forecasting process [1]. Various sections in literature review paper [1] describes need of Data Pool as a data structure.
Dhanashri V. Sahasrabuddhe and Dr. Pallavi P. Jamsandekar

Second section had explained study of different weather data formats used by different weather forecasting organizations. This study had helped researcher to understand need of design of Data Pool so that space (memory) used by the data can be optimized.

Third section gives design and algorithms for building, storing and accessing data from Data Pool. Fourth section explains about the implementation of the structure. The section talks about platform used for implementation as well as outputs for different files which vary in size. The structure is evaluated in fifth section. The section evaluates the structure for space usage for different size of data with same structure. To generalize usage of structure researcher had applied same structure to health data. Its results to health data are discussed in section six. The conclusion of the research work carried out by researcher is discussed in section seven.

2. Weather Data Formats and Functional dependency

Before applying any structure to data, the data needs to be understood. With the intention of applying some structure to weather data various study for data formats and weather forecasting process was carried out [2]. There are different formats, are categorized under three generic categories:

- **GRIB – Gridded Binary Data**
- **netCDF - Network Common Data Form or General Regularly-distributed Information in Binary form**
- **HDF - Hierarchical Data Format**

These formats are portable (machine independent) and self-describing. Following table (Table 1) describes these formats and their versions in details.

<table>
<thead>
<tr>
<th>Sr.No.</th>
<th>Format Name</th>
<th>Description</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GRIB1</td>
<td>GRIdded Binary (Edition 1)</td>
<td>World Meteorological Organization</td>
</tr>
<tr>
<td>2</td>
<td>GRIB2</td>
<td>GRIdded Binary (Edition 2)</td>
<td>World Meteorological Organization</td>
</tr>
<tr>
<td>3</td>
<td>netCDF3</td>
<td>Network Common Data Form (Version 3.x)</td>
<td>Unidata (UCAR/NCAR)</td>
</tr>
<tr>
<td>4</td>
<td>netCDF4</td>
<td>Network Common Data Format, (Version 4.x)</td>
<td>Unidata (UCAR/NCAR)</td>
</tr>
<tr>
<td>5</td>
<td>HDF4</td>
<td>Hierarchical Data Format, (Version 4.x)</td>
<td>NCSA/NASA</td>
</tr>
<tr>
<td>6</td>
<td>HDF4-EOS2</td>
<td>HDF4-Earth Observing System (Version 2 geo-referenced data)</td>
<td>--</td>
</tr>
<tr>
<td>7</td>
<td>HDF5</td>
<td>Hierarchical Data Format, (Version 5.x)</td>
<td>NCSA/NASA</td>
</tr>
<tr>
<td>8</td>
<td>HDF5-EOS5</td>
<td>HDF5-Earth Observing System, (Version 5 geo-referenced data)</td>
<td>--</td>
</tr>
<tr>
<td>9</td>
<td>GeoTIFF</td>
<td>Geo-referenced raster imagery</td>
<td>--</td>
</tr>
</tbody>
</table>

After studying different formats researcher come to the conclusion –
Data Pool: A Structure to Store Voluminous Data

- The data is in record format, depending on its category either stored as individual records or as file which is collection of records.
- Every record has predefined structure, which divides each record in fields
- Every field holds data of predefined data type
- Data is repetitive in nature
- These data formats helped researcher in understanding weather data format and to plan for design of structure.

3. Design and Algorithms for Data Pool
The conclusions about data, as stated in section 2, helped researchers to design the structure. With the objective of optimization of space, after concentrating on characteristics of data, researchers designed data structure which arranges data in two parts namely-
- Main or Base Structure
- Pool Structure
Which is collectively named as – ‘Data Pool’. These two parts are described in following two subsections.

3.1 Main or Base Structure
After observing weather data, and studying conclusions derived from study of data standard fields in any weather data table are –
- Station details,
- Date and time details for which data has been collected,
- Different weather parameters.

After observing data rows for the above fields, dependency among the fields can be defined. Dependency among attributes helps in defining hierarchical structure. For defining dependency, concept of functional dependency is used. Concept of Functional dependency is explained in the paragraph below.

Functional dependency is the term defined in relational database theory. In simple words the term Functional Dependency is defined as –

“When the value of one attribute (the determinant) determines the value of another attribute” [3]

A functional dependency allows us to express constraints that uniquely identify values of certain attributes with reference to value of a selected attribute. The concept is based on the concept of a relation defined in discrete mathematics. Relation R is defined between two sets written as X→Y. For two sets of attributes, set X in R is said to functionally determine another set of attributes Y if, and only if, each X value in R is associated with precisely one Y value in R; R is then said to satisfy the functional dependency X→Y. A functional dependency FD: X→Y is called trivial if Y is a subset of X.

For any format of weather data if we consider weather data table comprising of different columns as set X, then X as a set of attributes considering standard columns in weather data table can be defined as –

X= {Station details, Date, Time, Weather Parameters}
Y is defined as –

Y= {y | y∈X and Station Detail attribute value is common}
Y= {Date, Time, Weather Parameters}
While set Z is defined as –
\[ Z = \{ z \mid z \in Y \text{ and Date attribute value is common} \} \]
\[ Z = \{ \text{Time, Weather Parameters} \} \]
Here \( Z \subseteq Y \), then functional dependency between Y and Z defined as –
\[ \text{FD: } Y \rightarrow Z. \]
And as \( Y \subseteq X \), then functional dependency between X and Y defined as –
\[ \text{FD: } X \rightarrow Y. \]
Hence consequently by Transitivity rule functional dependency between X and Z can be defined as –
\[ \text{FD: } X \rightarrow Y \rightarrow Z. \]
This derivation of functional dependency among different attributes helps to define hierarchical structure to store data.

From the above defined functional dependency the hierarchies of the Base Structure are defined as –
- Weather parameter Name
- Station Details
- Date/Time
- Weather Parameter

### 3.2. Pool Structure

Use of only main or base structure for storing weather data is not sufficient for optimizing utilized space. With the intention of making it more efficient, another characteristic of the data was concentrated by researcher and that is repetitiveness.

While storing repeated data wastage of space can be observed. To avoid this wastage, instead of storing repeated data with main structure, researcher planned to store it separately using a pool. This pool goes on storing data without any context. In other words, data stored with the pool does not represent any specific field from weather data. It is just a collection of data items which are arranged properly so that insertion becomes easy. The advantage of keeping pool data context independent is any field can refer the same node if they are having same value.

For example if reading for temperature and precipitation are same both can refer to same node in pool.

Another advantage of keeping pool context independent is, same pool, as is a globally accessible to all different processes, can be shared by different main structures representing different weather parameters.

For example two simultaneous main structures representing different parameters like ‘temperature’ and ‘average temperature’ which are having different formats for base structure but can share same pool structure. Which helps to save more space.

### 3.3. Graphical presentation of structure

Pictorially the above discussed data structure along with it’s data arrangement is represented with the help of figure 1. The figure shows division of data in two parts Main structure and Pool structure.
Nodes in main structure are arranged in levels. Every node in any level have two connections, one connection is with next level, and another is with next node in the same level. Hence at each level a list of nodes is formed. Each node in list represents same field but different values. Root of the main structure defines weather parameter name, while level 1 defines nodes for storing station details, level 2 defines nodes for date and time details for its parent node (station) and the last level defines nodes for storing actual readings for parameters. These nodes in main structure does not store any data, but they store references to the nodes from data pool which stores actual data.

While adding data to Main structure, simultaneously Pool structure is built. When data value is read from file, firstly the value is added to Pool, and its reference is stored with node in Main structure.

4. OPERATIONS ON THE DATA POOL

For writing algorithms firstly it is necessary to understand hierarchies in Main structure, which is not possible without considering actual weather data fields. The data considered by researchers for implementation of the structure is used for writing algorithms. This data reads average temperatures for 129 years. First three columns describes station details for which data had been collected. While next 129 columns stores actual readings. Table 2 gives column headings for data while Table 3 gives sample data.

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Column Heading</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Elevation</td>
<td>Integer</td>
</tr>
<tr>
<td>2</td>
<td>UTMX</td>
<td>Double</td>
</tr>
<tr>
<td>3</td>
<td>UTMY</td>
<td>Double</td>
</tr>
<tr>
<td>4 - 131</td>
<td>Y.1,Y.2,.....Y.129</td>
<td>Double</td>
</tr>
</tbody>
</table>

Table 3. Sample data used for implementation
Dhanashri V. Sahasrabuddhe and Dr. Pallavi P. Jamsandekar

<table>
<thead>
<tr>
<th>Elev</th>
<th>UTMX</th>
<th>UTMY</th>
<th>y.1</th>
<th>y.2</th>
<th>y.3</th>
<th>y.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>288</td>
<td>5921725.464</td>
<td>2834892.009</td>
<td>-5.277777778</td>
<td>-1.666666667</td>
<td>4.833333333</td>
<td>7.833333333</td>
</tr>
<tr>
<td>25</td>
<td>5932444.735</td>
<td>2757031.495</td>
<td>-1.888888889</td>
<td>1.111111111</td>
<td>6.222222222</td>
<td>8.611111111</td>
</tr>
<tr>
<td>42</td>
<td>5946764.03</td>
<td>2852536.872</td>
<td>-4.444444444</td>
<td>-0.555555556</td>
<td>6.5</td>
<td>8.777777778</td>
</tr>
<tr>
<td>15</td>
<td>5785273.725</td>
<td>2462832.861</td>
<td>-0.277777778</td>
<td>2.555555556</td>
<td>8.5</td>
<td>11.055555556</td>
</tr>
<tr>
<td>254</td>
<td>4659138.411</td>
<td>2609427.39</td>
<td>-5.555555556</td>
<td>-0.833333333</td>
<td>6.277777778</td>
<td>8.277777778</td>
</tr>
</tbody>
</table>

From above data firstly hierarchies of the base structure needs to be defined. After functional dependencies among columns, the base structure was planned to have four levels. Every level in Base structure are explained as follows –

Level 0 –
For weather data always root node stores parameter name, for the considered data value assigned to parameter is ‘TAVG’.

Level 1 –
Next columns in table forms elevation details about the place for which data had been collected. As one elevation can have more than one UTMs, this level stores elevation data.

Level 2 –
Further for more station details X and Y co-ordinates were specified. This co-ordinates are dependent on elevation for specifying station details and hence at third level station specific data had been stored.

Level 3 –
Which is formed with leaf nodes for storing average temperature data for each year.

With level definitions design for Base structure was final. For data pool as the data is being read in text, it stores every data item as character array (string). By considering these structures for base and Data Pool the algorithms are discussed in next subsections.

4.1. Creating Main or Base Structure
Initially process starts from building main structure. Steps for structure building are given below

I. Initialize root to NULL
II. Accept Parameter and allocate root.
III. Read record from file.
IV. Read Elevation, if it is not previously added to structure at level 1, add node to Main structure.
V. Read UTM details for current elevation, if it is not previously added to structure at level 2, add node to main structure and add its values to Pool structure using insert Pool algorithm.
VI. Read year Data for current UTM, if it is not previously added to structure at level 3, add node to main structure and add its values to Pool Structure using insert Pool algorithm.
VII. Continue from step iii to vi till end of file has not reached.

4.2. Inserting to Main Structure
This algorithm helps user to insert new data from another file to existing structure. Limitation of the algorithm is, data with similar structure (columns) can only be added to the Main Structure.

i. Initialize cur to first elevation in level 1
ii. Read record from file.

iii. Read Elevation, if it is not previously added to structure at level 1, add node to Main structure.

iv. Read UTM details for current elevation, if it is not previously added to structure at level 2, add node to main structure and add its values to Pool structure using insert Pool algorithm.

v. Read year Data for current UTM, if it is not previously added to structure at level 3, add node to main structure and add its values to Pool Structure using insert Pool algorithm.

vi. Continue from step iii to vi till end of file has not reached.

4.3. Searching for Elevation
The algorithm is helpful when user wants to access or process data for particular elevation. Algorithm accepts elevation first and then start searching it at level1, when it is found data from next levels is accessed.

4.4. Searching for UTM
The algorithm is helpful when user wants to access or process data for particular elevation and UTM i.e data for particular station. When data for particular elevation is searched it may have more than one stations with same elevation. Algorithm accepts elevation and UTM co-ordinates for X and Y. Firstly it searches for elevation and then for UTM. When found it accesses data for the station.

5. IMPLEMENTATION AND EVALUATION OF DATA POOL STRUCTURE
The platform details used for implementing the structure are as follows –

Operating System : ubuntu 16.04 LTS (64 Bit)

Processor       : Intel core i5 CPU 2.00 GHz X 4
RAM       : 8GB

The structure was implemented using c language. The structure handles voluminous data and hence researcher has selected Ubuntu as operating system to execute the program. To understand the structure while execution small data was represented graphically which given in figure 1.

In figure 1 the upper part represents Main structure and the lower part represents Pool structure. Every node in Main structure displays reference to node in Pool.

While every node in Pool represents its own reference, for relevance of readers, and actual data value stored with node.

In figure it can be observed that the main structure is implemented as B+ tree, whose root is Parameter value. While nodes at every level forms linked list, which store unique value for the field.

The Pool structure is represented as BST (Binary Search Tree), which inserts value in pool after comparing value with parent node at each level.
Following subsection describes evaluation of Data Pool.

5.1. Evaluation of Data Pool for Weather Data
The data structure is evaluated by considering two of its characteristics –

- Space (Memory) Occupied by the structure while arranging data
- Time consumed (Comparisons and Iterations) while executing operations on data from structure.

Objective of the researcher was to design a data structure which can represent large data using optimum possible memory and helps to access data efficiently. Hence for evaluating the structure Space complexity and time complexity was calculated. The results of the evaluation are discussed in this section.

5.1.1. Space Complexity
For calculating space complexity for the structure, the structure was implemented for different size of data. While implementation six different attributes were applied which helped to carry out comparative study. The values for each attribute was calculated during implementation of structure, which are given in Table 4.
Table 4 Values for Different Attributes Considered for Evaluation of Weather Data Representation

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>File Name</th>
<th>netemp73.csv</th>
<th>netemp.csv</th>
<th>netempCopy.csv</th>
<th>ne1.csv</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>File Size (in KB)</td>
<td>72</td>
<td>511</td>
<td>1021</td>
<td>1447</td>
</tr>
<tr>
<td>2</td>
<td>No. of Records</td>
<td>50</td>
<td>356</td>
<td>712</td>
<td>1000</td>
</tr>
<tr>
<td>3</td>
<td>Main Structure (in KB)</td>
<td>49</td>
<td>346</td>
<td>345</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Pool Structure (in KB)</td>
<td>10</td>
<td>48</td>
<td>47</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>No. of Repetitions in Data</td>
<td>10%</td>
<td>15%</td>
<td>20%</td>
<td>60.00%</td>
</tr>
<tr>
<td>6</td>
<td>Percentage of reduction of data</td>
<td>45.83</td>
<td>41.68</td>
<td>70.81</td>
<td>99.59</td>
</tr>
</tbody>
</table>

Table 4 shows different attributes and their values which helped researcher to calculate space complexity. First attribute value shows size of memory occupied by original file data, while Main structure and Pool Structure collectively are memory occupied by data represented in Data Pool structure. The last attribute calculates ‘percentage of Reduction of Data’, which is calculated as –

\[ RD = \frac{(FS - DS)}{FS} \times 100 \]

Where -

RD – Reduction in Data Size
FS – File Size
DS – Total Memory Occupied by Structure (Main Structure + Pool Structure)

Graphically the data from Table 4 is represented as in Fig. 2. From graph and table it can be observed that as the size of data goes on increasing the percentage of reduction in data size goes on increasing. Also Percentage of reduction is directly proportionate to repetitions of data in original data, where space is saved while storing repetitive data.
After observing values in last row it can be observed that ‘there is comparative space reduction in size of memory used by the structure’. Hence it can be conclude that the structure is **Space Efficient**.

### 5.1.2. Time Complexity

While selecting data structure it time complexity is considered, which explains its time wise efficiency, the minimum time consumed by the structure in performing different operations on data stored with structure. Hence while calculating time complexity different algorithms are considered which are executed on data.

For the above implemented data structure as discussed earlier in section 4, four different algorithms were executed. As the data was comparatively large, to calculate time complexity for the structure, researcher concentrated on the basic structures used while implementation of Data Pool. The Table 5 represents time complexity for the structure –

**Table 5 Time Complexity of Structure**

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Algorithm name</th>
<th>Number of Items</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Building Main Structure</td>
<td>Level – m Node – n</td>
<td>O(m+n)</td>
</tr>
<tr>
<td>2.</td>
<td>Inserting to Main Structure</td>
<td>Level – m Node – n</td>
<td>O(m+n)</td>
</tr>
<tr>
<td>3.</td>
<td>Searching Elevation</td>
<td>Total elevations - n</td>
<td>O(n)</td>
</tr>
<tr>
<td>4.</td>
<td>Searching UTM</td>
<td>Total Elevations – m Total UTMs for Elevation - n</td>
<td>O(m+n)</td>
</tr>
<tr>
<td>5.</td>
<td>Insert in Pool</td>
<td>Number of Levels - n</td>
<td>O(n)</td>
</tr>
<tr>
<td>6.</td>
<td>Searching node using node number in Pool</td>
<td>Number of Levels - n</td>
<td>O(n)</td>
</tr>
</tbody>
</table>

**Figure 2** Graphical representation for Comparative Study for Weather Data

![Graphical representation for Comparative Study for Weather Data](image)

![File Size, Main Structure, No. of Repetitions in Data, No. of Records, Pool Structure, Percentage of reduction of data](image)
The Table explains time complexity for the structure which shows, as the Data Pool applies ‘Divide and Conquer’ technique on data, comparatively it consumes less time in searching for data.

5.1.3. Data Loss

Today data has become more precious and hence while transforming data from file to any structure significance of data loss needs to be considered. Hence with the intension of testing for any data loss, researchers have carried out ANOVA (Analysis of Variance) test. Hypothesis for calculating F statistic were defined as –

\[ H_0 : \text{There is no significant difference in original data and Data represented in Pool structure.} \]

\[ H_1 : \text{There is significant difference in original data and Data represented in Pool structure.} \]

For the purpose of testing hypothesis researcher selected a file named ‘netemp.csv’ with 356 records. Each record in file represents average temperature for a particular area identified uniquely by elevation, UTMX and UTMY. The test was required to be carried out on data for selected area having 129 fields for years. So it was One-Way Classified Data. For calculating F parameter value following steps are followed –

I. Accept station details from user for which data needs to be tested.

II. From file select the record for the area and to CSV file.

III. For the same area using ‘searchUTM’ algorithm add year’s data to the same CSV file as a next record.

IV. Open newly created CSV file using Excel and apply f-test function on the year’s data (exclude station data).

V. After applying function it returns value of parameter ‘F’.

VI. Compare the value with tabulated value of ‘F’ by considering level of significance (1\% and 5\%) and degrees of freedom are defined as –

\[ \text{VII. Numerator } = \text{Number of Groups (k) } - 1 \]

\[ \text{VIII. Denominator } = \text{Total Number of Observations(N) } - k \]

IX. To conclude about the test – if tabulated value is greater than calculated value accept the null hypothesis otherwise reject it.

Researcher has calculated value of ‘F’ for all records from file. The calculated value of ‘F’ for each record was ‘1’.

When data from two or more groups is exactly same calculated value of F-Statistic is 1. Hence the conclusion of the test was –

**Accept the null hypothesis, i.e. there is no significant difference in original data and data represented in Pool structure**

6. APPLICATION AND EVALUATION OF DATA POOL FOR HEALTH DATA

To generalize the usage of the Data Pool structure, researcher implemented the same for health data. Health data is also one of the prominent source of big data which is considered to be precious with consideration of social value.

The structure of the implemented health data is described in Table 6.

**Table 6 Data Format for Health Data**
After observing file structure and defining functional dependency among columns, levels in Main Structure were defined as –

- Level 0 – Indicator category
- Level 1 – Indicator, Shortened Indicator Name, Shortened Indicator Name (Graph)
- Level 2 – Year
- Level 3 – Sex, Race / Ethnicity
- Level 4 – Value, Place, BCHC Requested Methodology, Source, Methods, Notes, 95% Confidence Level – Low, 95% Confidence Level – High

Graphically, after implementing the structure, its output is represented in Fig.3.
Table 6.2 Values for Different Attributes Considered for Evaluation of Health Data Representation

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>File Name</th>
<th>File Size (in KB)</th>
<th>No. of Records</th>
<th>Main Structure (in KB)</th>
<th>Pool Structure (in KB)</th>
<th>No. of Repetitions in Data</th>
<th>Percentage of reduction of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hd100.csv</td>
<td>56.45</td>
<td>100</td>
<td>12.49</td>
<td>12.33</td>
<td>17.67</td>
<td>99.71</td>
</tr>
<tr>
<td>2</td>
<td>hd200.csv</td>
<td>101.04</td>
<td>200</td>
<td>24.27</td>
<td>19.29</td>
<td>17.63</td>
<td>95.06</td>
</tr>
<tr>
<td>3</td>
<td>hd200r.csv</td>
<td>89.92</td>
<td>200</td>
<td>7.58</td>
<td>3.67</td>
<td>28.83</td>
<td>95.65</td>
</tr>
<tr>
<td>4</td>
<td>hd300.csv</td>
<td>145.67</td>
<td>300</td>
<td>35.90</td>
<td>24.41</td>
<td>17.58</td>
<td>92.11</td>
</tr>
<tr>
<td>5</td>
<td>hd300r.csv</td>
<td>135.22</td>
<td>300</td>
<td>2.23</td>
<td>1.67</td>
<td>32.33</td>
<td>99.59</td>
</tr>
<tr>
<td>6</td>
<td>healthData.csv</td>
<td>8165.22</td>
<td>18950</td>
<td>1493.94</td>
<td>433.90</td>
<td>25.17</td>
<td>87.02</td>
</tr>
</tbody>
</table>

After observing the values for attributes for six different file size the structure was found to be more space efficient as compared to weather data. This is due to the size of data values hold by each column for both the data. Weather data defines 132 columns. But size of data hold by each column is maximum up to 12 characters. And hence depending on number of columns overhead used increases, but size of reference value hold by node in main structure is same as size of data value hold by node in pool. This characteristic, to some extent, reduces effectiveness of application of structure for weather data, as compared to health data.
7. CONCLUSION
Through current research work, researcher had designed and developed a new data structure which can be applied to any structured big data while carrying out any process. Characteristics of the outcome of the research are stated as –

- **Reduction in size** –
  
  From the calculated space complexity for the data, reduction in size can be observed. Current era is of ‘data flood’, and if the structure helps to reduce size, for a process maximum data can be available.

- **Reduction in execution Time** –
  
  Time complexity for the structure shows that, the structure adds efficiency to searching process, as it follows the technique of ‘Divide and Conquer’. Instead of searching from a file which is large in file, it becomes efficient using structure.

- **Loss Less Representation** -
  
  With the test it is proved that, there is no data loss using structure. Process can get same data as original. Today data is more precious, and any business or process cannot afford data loss, which affects results.

  With reference to discussed research work, its outcome is a new innovative data structure named ‘Data Pool’. This is the era of big data. Big data is the area where maximum innovations are taking place. All industries today are facing problem of big data. And at the moment the structure will help to find some solution to the problems faced.

  This research has not considered segmentation which helps to represent data by parts. Currently researcher has considered representation of entire data at a particular instance of time.

REFERENCES

