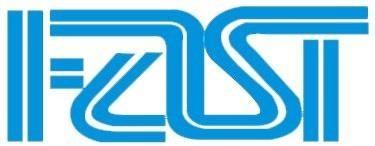
***National University of Computer &***

***Emerging Sciences Karachi Campus***



MULTI ATTRIBUTE AVL TREE ADAPTER (M.A.A.T.A)

***Group Members***

1. **Shahzaib Khan (19k-0273)**
2. **M. Uzair Abbasi (21k-4548)**
3. **Bhavish Kumar (21k-3450)**

# **PROBLEM STATEMENT**

# In single (key) attributed trees (BST/AVL etc), we usually point data of one type at a time (i.e. we can use comparator for only one attribute), but the problem occurs when we need trees for storing structure of multiple attributes & require optimize results for all attributes. Means, for AVL, it can then only give optimized results for one attribute i.e. O(logN) while for rest its O(N) time-complexity.

# **PROPOSED SOLUTION**

Instead of Making N trees for N attributes, we will use address book keeping technique & save references to one-time stored object’s respective attribute. Theoretically similar to:

**AVL<Pair<RespectiveType\*, List<RecordType\*>>> Trees[TotalAttributes];**

Here, each AVL tree individually points to a specific attribute of record & also to a list of pointers to the duplicates of that record.

[refer to concept diagram for better understanding]

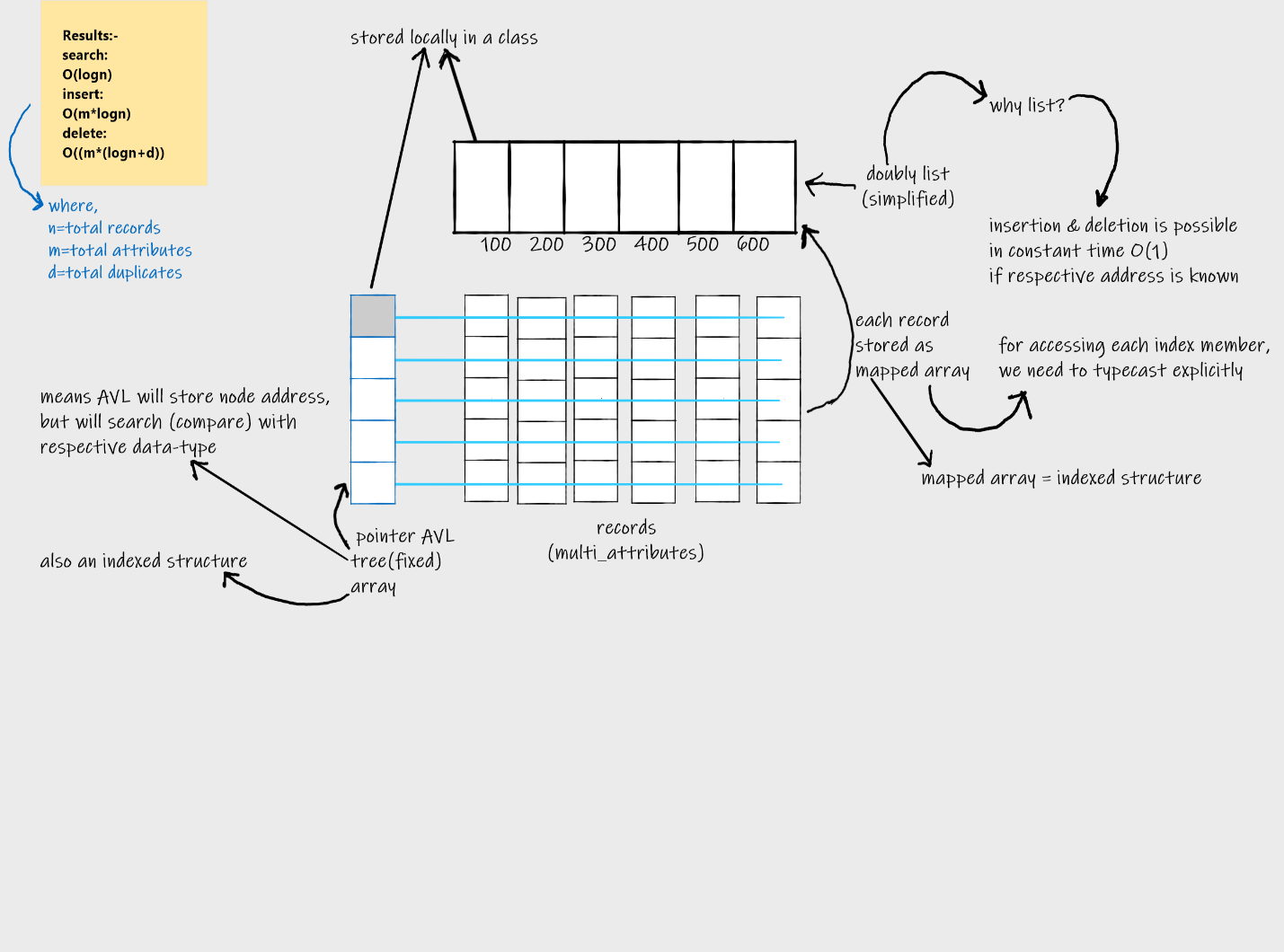
**SALIENT FEATURES**

* It can be used for any structure/object.
* It will handle duplicates too.
* Will Provide filter search & delete operations.
* Read/Write/Overwrite complete .txt or .csv files after/before applying M.A.A.T.A operations.

**APPLICATION**

We will use M.A.A.T.A for data manipulation in Excel files.

**SIMPLIFIED CONCEPT DIAGRAM**



**TOOLS & TECHNOLOGIES**

1. DevC++ compiler.
2. C++11 programming language.
3. Operating System Microsoft Windows 10/11.