

DBMS PROJECT

DATA STRUCTURES

SECTION: E

GROUP MEMBERS:

Mohammad Attique 21i-0410

Saif Ali Khan 21i-0649

Abdullah Khan 21i-0805

Each member played a vital role in implemented and understanding the project. All member equally participated in completing the project. RED BLACK TREES was done by Mohammad Attique, AVL TREES was implemented by Saif Ali Khan and B-TREES was implemented by Abdullah Khan. Although each member focused on one tree but everyone contributed in helping and understanding the depth of project.

Separate nodes are made for each link list and each tree which made work easier. Templates were also used effectively which helped in avoiding making of Different function for inserting deletion and searching. One Template key was passed which completed the purpose of indexing of trees.

RED BLACK TREES:

Red black trees node consists of every data including id, death, years, cause , ADR etc.

REDBLACK trees are basically AVL trees with maximum 2 rotation and slightly blanaced. REDBLACK trees insertion was done simple BST insertion and keeping in mind REDBLACK TREE properties. Root was inserting as black and every other new node was inserted as red. If parent was red and child was color was also red, there was a red, red conflict then color of parent's sibling was checked if it was Red then it was Recolor, else if it is colored Black or NULL then make suitable Rotation were done and Recolored it. This process was Repeated the same until tree becomes Red Black Tree.

Deletion was likewise completed using a straightforward BST deletion, but we must also examine the Red-Black Tree features. Make the appropriate operations, such as recoloring, rotating, and recoloring after rotation, if any of the properties are violated. We rotate if a sibling was black and one or more of their offspring were red. We did recoloring and repeated it for the parent if the parent was black if the sibling was black and both of its offspring were black. If a sibling was red, we rotated them to shift the older sibling up and changed the colour of the father and older sibling. The new child is invariably black.

B TREES:

B Trees insertion is somewhat similar to BST but it just contains different data in its node. E.g if 4 way B tree node will have 3 values lets take 5,10 and 15 respectively. Values less than 5 will be inserted before 5 values greater than 5 and less than 10 will be inserted between 5 and 10, vals greater than 10 and less then 15 b/w 10 and 15 and vals greater than 15 after 15. Keeping in mind other several properties of B_trees insertion and deletion were done.

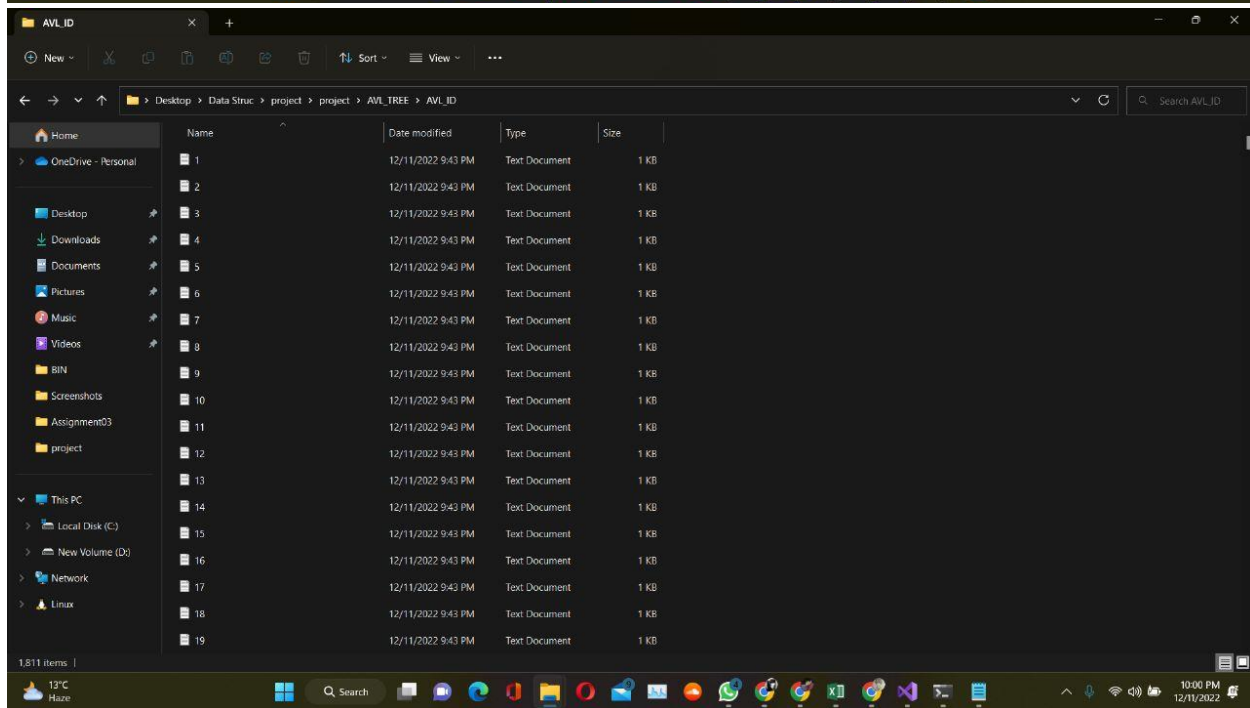
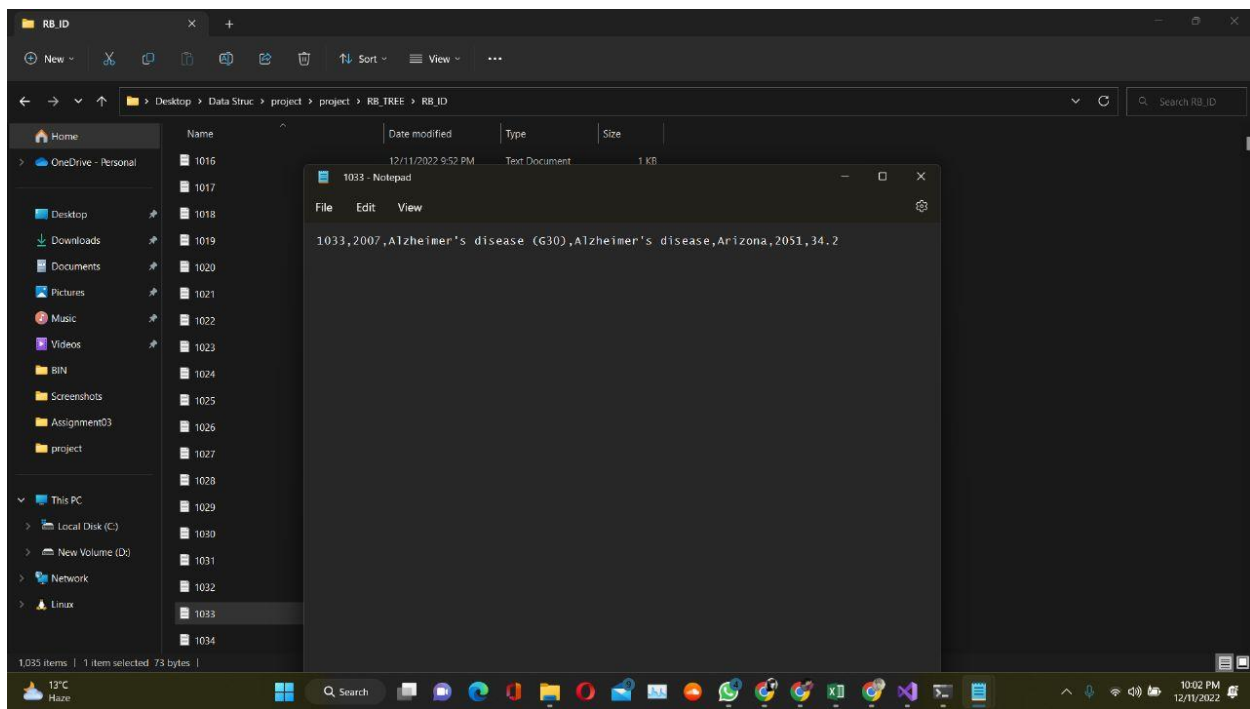
AVL- trees:

The definition of an AVL tree is a height-balanced binary search tree in which each node has a balance factor that is determined by deducting the height of the node's right subtree from the height of its left subtree. Going up the recursive call, my autobalancing function checks each node's balance factor to see if it is more than or equal to two. it will perform necessary rotations, i.e single or double rotation Templates Helped us in making an AVL tree for different Data types

HEAP:

Files were read once and were stored in trees later on these trees were stored on hdd for quicker access using multiple libraries. This allowed dbms system to run faster and efficiently. Each tree was stored in different directory having different directory relatind to its index searching sample is shown below

Major problems that occurred were during understanding how to deal with the project, and implementation of trees and was the time management since we had other assignment pending too. But by taking help of each other during difficult times we were able to tackle such problems allowing us to learn and improve more






















RB_CAUSE_NAME	12/11/2022 9:46 PM	File folder
RB_DEATHS	12/11/2022 9:40 PM	File folder
RB_ID	12/11/2022 8:44 PM	File folder
RB_YEAR	12/11/2022 8:22 PM	File folder

```

42 template <typename T>
43 struct RBNode
44 {
45     T key;
46     int id;
47     int years;
48     int line_no;
49     string Cause;
50     string name;
51     string state;
52     int deaths;
53     double ADR;
54     List<T> duplicate;
55
56     RBNode<T>* left = nullptr;
57     RBNode<T>* right = nullptr;
58     RBNode<T>* parent = nullptr;
59     bool red = 0;
60     bool black = 1;
61     int height;
62
63     RBNode(T key=0, int id = 0, int yrs = 0, string cuz="", string name="", string state="", int dths=0, double ADR=0, int line=0)
64     {
65         this->key = key;
66         this->id = id;
67         years = yrs;
68         Cause = cuz;
69         this->name = name;
70         this->state = state;
71         deaths = dths;
72         this->ADR = ADR;
73         this->line_no = line;
74         height = 1;
75     }

```

 AVL_TREE	12/11/2022 9:40 PM	File folder	
 project	12/11/2022 9:46 PM	File folder	
 RB_TREE	12/11/2022 9:46 PM	File folder	
 x64	12/6/2022 11:15 PM	File folder	
 AVL_Tree.h	12/11/2022 9:42 PM	C/C++ Header	28 KB
 LIST.h	12/11/2022 9:42 PM	C/C++ Header	3 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	57 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	77 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	75 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	70 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	79 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	90 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	99 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	111 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	99 KB
 NCHS_-_Leading_Causes_of_Death_Unit...	12/10/2022 3:59 PM	Microsoft Excel Co...	85 KB
 project.zip	12/11/2022 9:47 PM	Compressed (zipp...	14 KB
 Projectnew.vcxproj	12/11/2022 4:18 PM	VC++ Project	7 KB
 Projectnew.vcxproj.filters	12/11/2022 4:18 PM	VC++ Project Filte...	2 KB