National University of Computing And Emerging Sciences

Report of Data Structure

Prepared by:

Aqeel Afzal # 19I-0650

Abid Hussain # 19I-1982

Sahil Parkash#19I-0679

Section:

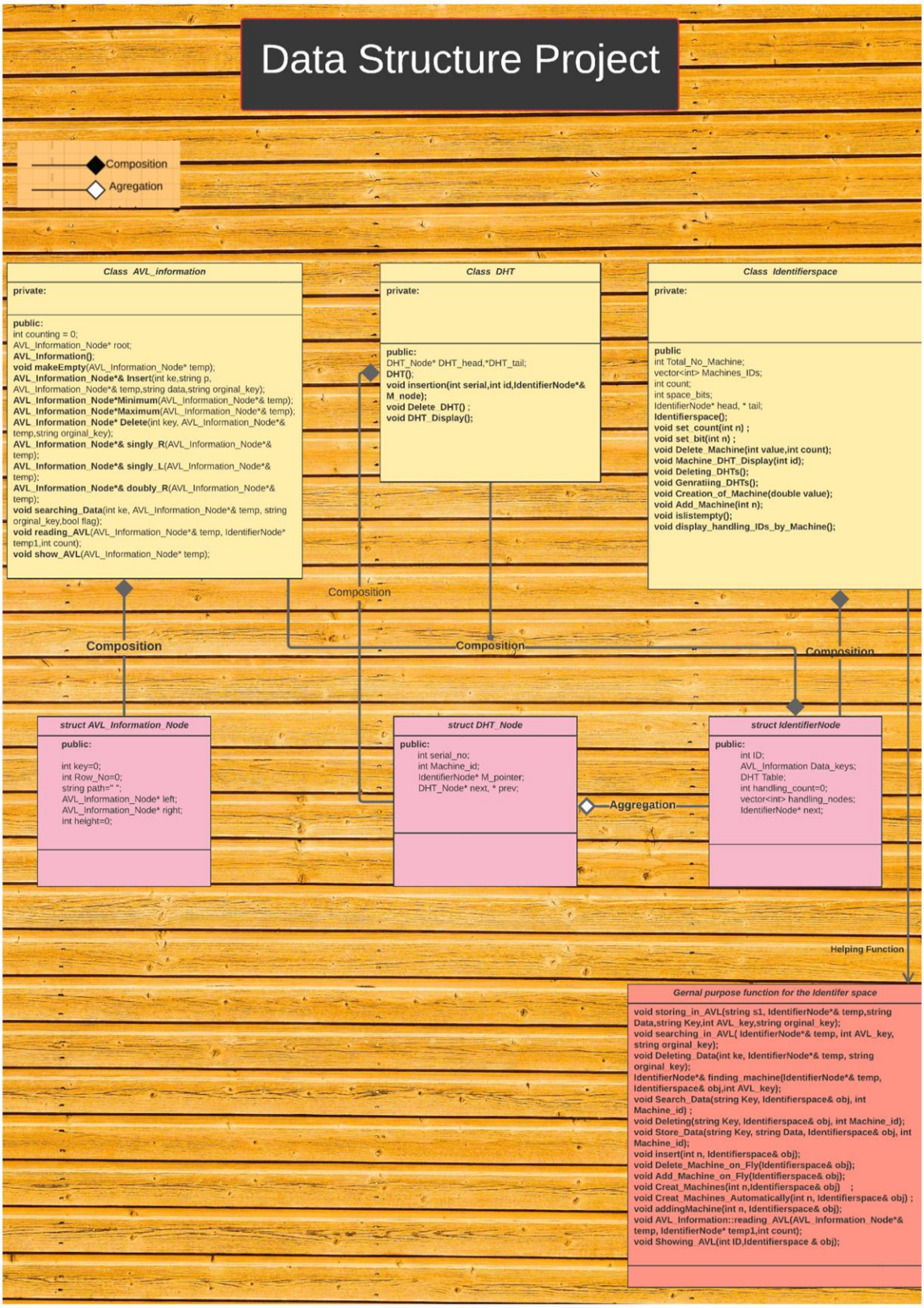
CS-D

Submitted to:

Ma’am Saba Rashid

Submitted on:

27/12/2020


Data Structure Project
Composition
Agregation
Class AVL_information
private:
public:
int counting = 0;
AVL_Information_Node* root;
AVL_Information();
void makeEmpty(AVL_Information_Node* temp);
AVL_Information_Node*& Insert(int ke,string p,
AVL_Information_Node*& temp,string data,string orginal_key);
AVL_Information_Node*Minimum(AVL_Information_Node*& temp);
AVL_Information_Node*Maximum(AVL_Information_Node*& temp);
AVL_Information_Node* Delete(int key, AVL_Information_Node*&
temp,string orginal_key);
AVL_Information_Node*& singly_R(AVL_Information_Node*&
temp);
AVL_Information_Node*& singly_L(AVL_Information_Node*&
temp);
AVL_Information_Node*& doubly_R(AVL_Information_Node*&
temp);
void searching_Data(int ke, AVL_Information_Node*& temp, string
orginal_key,bool flag);
void reading_AVL(AVL_Information_Node*& temp, IdentifierNode*
temp1,int count);
void show_AVL(AVL_Information_Node* temp);
Class DHT
private:
public:
DHT_Node* DHT_head,*DHT_tail;
DHT();
void insertion(int serial,int id,IdentifierNode*&
M_node);
void Delete_DHT() ;
void DHT_Display();
Class Identifierspace
private:
public
int Total_No_Machine;
vector<int> Machines_IDs;
int count;
int space_bits;
IdentifierNode* head, * tail;
Identifierspace();
void set_count(int n) ;
void set_bit(int n) ;
void Delete_Machine(int value,int count);
void Machine_DHT_Display(int id);
void Deleting_DHTs();
void Genratiing_DHTs();
void Creation_of_Machine(double value);
void Add_Machine(int n);
void islistempty();
void display_handling_IDs_by_Machine();
Composition
Composition
Composition
Composition
struct AVL_Information_Node
public:
int key=0;
int Row_No=0;
string path=" ";
AVL_Information_Node* left;
AVL_Information_Node* right;
int height=0;
struct DHT_Node
public:
int serial_no;
int Machine_id;
IdentifierNode* M_pointer;
DHT_Node* next, * prev;
Aggregation
struct IdentifierNode
public:
int ID;
AVL_Information Data_keys;
DHT Table;
int handling_count=0;
vector<int> handling_nodes;
IdentifierNode* next;
Helping Function
Gernal purpose function for the Identifer space
void storing_in_AVL(string s1, IdentifierNode*& temp,string
Data,string Key,int AVL_key,string orginal_key);
void searching_in_AVL(IdentifierNode*& temp, int AVL_key,
string orginal_key);
void Deleting_Data(int ke, IdentifierNode*& temp, string
orginal_key);
IdentifierNode*& finding_machine(IdentifierNode*& temp,
Identifierspace& obj,int AVL_key);
void Search_Data(string Key, Identifierspace& obj, int
Machine_id) ;
void Deleting(string Key, Identifierspace& obj, int Machine_id);
void Store_Data(string Key, string Data, Identifierspace& obj, int
Machine_id);
void insert(int n, Identifierspace& obj);
void Delete_Machine_on_Fly(Identifierspace& obj);
void Add_Machine_on_Fly(Identifierspace& obj);
void Creat_Machines(int n,Identifierspace& obj) ;
void Creat_Machines_Automatically(int n, Identifierspace& obj) ;
void addingMachine(int n, Identifierspace& obj);
void AVL_Information::reading_AVL(AVL_Information_Node*&
temp, IdentifierNode* temp1,int count);
void Showing_AVL(int ID,Identifierspace & obj);

Report of Data Structure Project

struct IdentifierNode

These are the data members

- int ID;
  - ⇨ used to store the ID of a machine.
- AVL_Information Data_keys;
  - ⇨ Used to store the key of data.
- DHT Table;
  - ⇨ Used to provide DHT table to a particular machine.
- vector<int> handling_nodes;
  - ⇨ used to store all identifiers nodes handle by a particular machine.
- IdentifierNode* next;
  - ⇨ Used to move to next node.

Class Identifierspace

Purpose:

It is used to create the identifier spaces, which depends upon the number of bits, and this will create the singly circular link list of the machines. Which means this class is responsible for creating, deleting, and adding a machine in the identifier space.

private:

- ⇨ no private data members.

public

These are the data members

- ➢ vector<int> Machines_IDs;
  - ⇨ used to store all machines IDs.
- ➢ int count;
  - ⇨ Total places in space.
- ➢ int space_bits;
  - ⇨ Total bits in space.
- ➢ IdentifierNode* head, * tail;
  - ⇨ Used to move the at head and tail.
- ➢ Identifierspace();
  - ⇨ Default constructor.
- ➢ int Total_No_Machine;
  - ⇨ used to store the total number of machines.

These are the Function

- void set_count(int n) ;
  ⇨ used for total places in space.
- void set_bit(int n) ;
  ⇨ used for total bits in space.
- void Delete_Machine(int value,int count);
  ⇨ used for deleting a machine.
- void Machine_DHT_Display(int id);
  ⇨ Printing the DHT table of a particular machine.
- void Deleting_DHTs();
  ⇨ used to delete the DHT table.
- void Genratiing_DHTs();
  ⇨ used to Genration of DHTs of all the machines
- void Creation_of_Machine(double value);
  ⇨ used to create the machine in the identifier space
- void Add_Machine(int n);
  ⇨ used to adding machine
- void islistempty();
  ⇨ used to simply check weather link list is empty or not.
- void display_handling_IDs_by_Machine();
  ⇨ used to displaying all the space handled by a particular machine

struct DHT_Node

These are the data members

- int serial_no;
  ⇨ index number for DHT table.
- int Machine_id;
  ⇨ used for machine ID.
- IdentifierNode* M_pointer;
  ⇨ IdentifierNode pointer used here for moving in the identifier node of a machine in order to direct traversing.
- DHT_Node* next, * prev;
  ⇨ Used for traversing in the DHT nodes.

Class DHT

Purpose:

It is used to create the DHT table for machines, which depend upon the number of space bit. Which means this class is responsible for creating, assigning and deleting a DHT table for a machine.

private:

⇨ no private data members.

These are the Functions

public:

- DHT_Node* DHT_head,*DHT_tail;
  - ⇨ DHT nodes in order to move in DHT nodes.
- DHT();
  - ⇨ Default constructor
- void insertion(int serial,int id,IdentifierNode*&M_node);
  - ⇨ used to inserting information in DHT
- void Delete_DHT() ;
  - ⇨ used to Deleting DHTs
- void DHT_Display();
  - ⇨ Used for Displaying the DHT table.

struct AVL_Information_Node

public:

These are the data members

- int key=0;
  - ⇨ used to store the key.
- int Row_No=0;
  - ⇨ used to store the particular line number of the txt file.
- string path=" ";
  - ⇨ used to store the path of a file.
- AVL_Information_Node* left;
  - ⇨ Used to move in the left direction of the AVL tree
- AVL_Information_Node* right;
  - ⇨ Used to move in the left direction of the AVL tree
- int height=0;
  - ⇨ used to store the height of overall AVL tree.

Class AVL_information

Purpose:

It is used to store All data entered by the user and contain address of all data. Which means this class controls the data stores, delete, and add by creating the txt files for a particular machine.

private:

- ⇨ no private data members.

public:

These are the Functions

- int counting = 0;

⇨ used for counting purpose, how many records are stored in the file.

- AVL_Information_Node* root;
  ⇨ Root of AVL tree.
- AVL_Information();
  ⇨ Default constructor.
- void makeEmpty(AVL_Information_Node* temp);
  ⇨ used to to delete the tree.
- AVL_Information_Node*& Insert(int ke,string p, AVL_Information_Node*& temp,string data,string orginal_key);
  ⇨ Used to insert the to insert the data
- AVL_Information_Node*Minimum(AVL_Information_Node*& temp);
  ⇨ Used to find minimum in tree.
- AVL_Information_Node*Maximum(AVL_Information_Node*& temp);
  ⇨ Used to find maximum in tree
- AVL_Information_Node* Delete(int key, AVL_Information_Node*& temp,string orginal_key);
  ⇨ Used to delete a particular data in AVL tree
- AVL_Information_Node*& singly_R(AVL_Information_Node*&temp);
  ⇨ Used for right shift
- AVL_Information_Node*& singly_L(AVL_Information_Node*&temp);
  ⇨ Used for left shift
- AVL_Information_Node*& doubly_R(AVL_Information_Node*& temp);
  ⇨ Used to traversing in the right direction of the tree.

- AVL_Information_Node*& doubly_L(AVL_Information_Node*& temp);
  ⇨ Used to traversing in the right direction of the tree.
- void searching_Data(int ke, AVL_Information_Node*& temp, string orginal_key,bool flag);
  ⇨ used to search data using its particular key.
- void reading_AVL(AVL_Information_Node*& temp, IdentifierNode* temp1,int count);
  ⇨ not define yet……
- void show_AVL(AVL_Information_Node* temp);
  ⇨ used to print the AVL of a particular

Gernal purpose function for the Identifer space

<u>These are the Genrnal Function:</u>

These functions are used for helping our already created classes.

- void storing_in_AVL(string s1, IdentifierNode*& temp,string Data,string Key,int AVL_key,string orginal_key);

- void searching_in_AVL(IdentifierNode*& temp, int AVL_key, string orginal_key);
- void Deleting_Data(int ke, IdentifierNode*& temp, string orginal_key);
- IdentifierNode*& finding_machine(IdentifierNode*& temp, Identifierspace& obj,int AVL_key);
- void Search_Data(string Key, Identifierspace& obj, int Machine_id);
- void Deleting(string Key, Identifierspace& obj, int Machine_id);
- void Store_Data(string Key, string Data, Identifierspace& obj, int Machine_id);
- void insert(int n, Identifierspace& obj);
- void Delete_Machine_on_Fly(Identifierspace& obj);
- void Add_Machine_on_Fly(Identifierspace& obj);
- void Creat_Machines(int n,Identifierspace& obj) ;
- void Creat_Machines_Automatically(int n, Identifierspace& obj) ;
- void addingMachine(int n, Identifierspace& obj);
- void AVL_Information::reading_AVL(AVL_Information_Node*&temp, IdentifierNode* temp1,int count);
- void Showing_AVL(int ID,Identifierspace & obj);