**“Ring DHT Explanatory Document”**

* **Relation between classes:**
* **hashh** has a direct relation with **machines**.
* **machines** class have direct relation with **Routing Table** class and **AVL** class.
* **Routing Table** and **AVL** classes don’t further have class relations inside them.
* **Working of Classes:**
* **hashh** class would be our main class from which we will make further relations.

It could be seen as the class that creates the Ring DHT.

* **machines** class creates the circular singly linked list which is the ring of machines in identifier space.

Thus, each machine creates a routing table and an AVL tree for itself.

* While **Routing Table** class will create a doubly linked list which will have a doubly link list of each machines Routing Table.
* The **AVL** class will create an AVL tree which will be present in every machine.
* **Functionality:**

**Main:**

* The main function is basically the driving function in whole project.
* First and fore most our program asks for the number of bits on which Ring DHT is to be implemented while creating the corresponding identifier space.
* And then it takes the number of specific/responsible machines that are to be used in out identifier space.
* After that we have implemented a menu-based program for all the required functionalities for a better user interference.

**hashh:**

All our different functionalities that were required from us in this project are implemented in this class in each function.

* **Add\_machines()** : It populates/adds machine in the machines ring in identifiers space.
* **Insert\_Key():** It inserts Data ID which stores value in AVL of that respective Machine which is responsible for this DATA ID.
* **Look\_up\_RT() :** This is the function that is used as a search algorithm for efficiently traversing within the ring between machines.
* **Hash()**: This function is used as a Hashing function which will hash the inputs given by the user .
* **Del\_machines():** It deletes machine in the machines ring in identifiers space.

**Machines**:

Machines class will be dealing with the functions that will be related to the functionality of machines in the identifiers space. Its addition, deletion, display, and other inter related functionalities are done here.

**Routing Table:**

This class is for making routing tables for all the machines. As these routing tables will help in search algorithm. There resettling and other functionalities are done in this class.

**AVL:**

AVL class is making trees on each machine that will store the values that are inserted in the nodes that this machine is responsible of. Also, it is dealing with file reading and writing inside its functions.