**Relationship between different classes and further explanation**

1. **CircularLinkedList:**

**Relation*:***

* CircularLinkedList contains CircularNodes, representing a composition relationship.
* Utilizes BigInt for various operations, implying a dependency relationship.
* Internally uses BTreeNodes through CircularNodes, indicating an association.

**Functions:**

* BigInt totalSize; - Represents the total size of the CircularLinkedList.
* int logSize; - Represents the logical size of the CircularLinkedList.
* CircularNode\* root; - Points to the root CircularNode of the CircularLinkedList.
* CircularLinkedList(int identifier) - Constructor for CircularLinkedList.
* void displaymybtree(BigInt i) - Displays the BTree associated with a specific BigInt.
* void removefile(int hval) - Removes a file based on its hash value.
* string search(int hval) - Searches for a file based on its hash value.
* void addfile(const string& path) - Adds a file to the CircularLinkedList.
* void insertion(BigInt id) - Inserts a CircularNode with the given BigInt.
* bool DeleteDirectory(const std::wstring& path) - Deletes a directory.
* void deletion(BigInt id) - Deletes a CircularNode with the given BigInt.
* void display() - Displays the CircularLinkedList.
* void updateRoutingTable() - Updates the routing table.
* void displayRoutingTable() - Displays the routing table.
* CircularNode\* traverseRoutingTable(BigInt startID, BigInt machineID) - Traverses the routing table.

1. **Bigint:**

**Relation:**

* BigInt is used within CircularLinkedList and CircularNode for various operations, signifying a dependency relationship.

**Functions:**

* Various arithmetic, assignment, and logical operators for BigInt manipulation.
* Constructors for initializing BigInt from int, string, or another BigInt.
* Conversion operators for converting BigInt to string or int.
* Overloaded input and output operators for streaming BigInt values.

1. **CircularNode:**

**Relation:**

* CircularNode is part of CircularLinkedList, indicating a composition relationship.
* Contains a pointer to DoublyLinkedList, establishing an association.
* Holds a BTreeNode, implying an association with the BTreeNode class.

**Functions:**

* BigInt ID; - Represents the identifier of the CircularNode.
* int size; - Represents the size of the CircularNode.
* BigInt tSize; - Represents the total size of the CircularNode.
* DoublyLinkedList\* T; - Points to the DoublyLinkedList associated with the CircularNode.
* CircularNode\* next; - Points to the next CircularNode in the CircularLinkedList.
* BTreeNode\* btreeroot; - Points to the root of the BTree associated with the CircularNode.
* CircularNode(BigInt i, int totalSize) - Constructor for CircularNode.
* void update() - Updates the CircularNode.
* void display() - Displays the CircularNode.
* CircularNode\* traverseRoutingTable(BigInt machineID) - Traverses the routing table.

1. **BTreeNode:**

**Relation:**

* BTreeNode instances are contained within CircularNode, representing a composition relationship.

**Functions:**

* Various functions for insertion, deletion, searching, and manipulation of BTree nodes.
* displayb(BTreeNode\*) - Displays the BTree.
* print\_tree(BTreeNode\* root, int prelen) - Prints the BTree in a tree-like format.

1. **DoublyLinkedList:**

**Relation:**

* CircularNode has a pointer to DoublyLinkedList, indicating an association

**Functions:**

* Node\* head; - Points to the head of the DoublyLinkedList.
* Node\* tail; - Points to the tail of the DoublyLinkedList.
* Node\* search(BigInt d) - Searches for a node with the given BigInt in the DoublyLinkedList.
* void insertAtEnd(BigInt d, CircularNode\* C) - Inserts a node at the end of the DoublyLinkedList.
* void insertAfter(BigInt after, BigInt newData, CircularNode\* C) - Inserts a node after a specified node in the DoublyLinkedList.
* void display() - Displays the DoublyLinkedList.
* void removeNode(BigInt d) - Removes a node with the given BigInt from the DoublyLinkedList.

1. **Node:**

**Relation:**

* Node is part of DoublyLinkedList, indicating a composition relationship.
* Points to a CircularNode, establishing an association.

**explanation:**

CircularNode\* address; - Points to the CircularNode associated with the Node.

Node\* next; - Points to the next node in the DoublyLinkedList.

Node\* prev; - Points to the previous node in the DoublyLinkedList.