

COP 5536 SPRING 2019 PROGRAMMING PROJECT

Implementation of B+ trees using JAVA

Name: Bing Qi
UFID: 32273501
Email: bing.qi@ufl.edu

This programs have six main parts:

- 1) Tree.java
- 2) bplustree.java
- 3) Node.java
- 4) IndexNode.java
- 5) LeafNode.java
- 6) Key.java

File include:

- 1) input.txt

1. bplustree.java

This is the main part of the whole project. The functions of this java file include:

- 1) PrintResult (Key) and PrintResult(ArrayList<Key> keys)

These two functions have the purpose to transfer the search functions of the node into string formats that can be written into the out.put files

- 2) main functions

the main parts of the main functions is to read each lines from the input file, and execute the functions from the file include initialize the trees, insert the index, search the index or index ranges, and remove the index from the tree. Then, write the results of the search and range search into the output.txt file.

2. Tree.java

This is to construct a B+ Tree. And the input parameters to build up the tree include:

- 1) order. This is the order of the tree

The variable of the tree include:

- 1) root. This is the root of the tree

The constructor of the tree class is the Tree (int treeOrder)

The functions of the class include:

Function	Type	Input	Return
insertIntoTree	Node	key of the node	root of the tree
searchTree	Key	index of the key	value of the key
searchTree	ArrayList<Key>	index range	value list of the key
remove	Void	index of the key	Null

3. Node.java

This is an abstract class and will be extended by index node and leaf node. Some functions shared by both these two other nodes will be written here

The variable of the class include:

- 1) keys. The keys list inside a node
- 2) parent. The parent of the node
- 3) maxsize. The maximum size of the keys inside a node.
- 4) children. The children list of the node.

Some function of the Node:

Function	Type	Input	Return
setParent	Void	indexNode	Void
getKeys	LinkedList	Null	Keys of node
getMaxSize	Int	Null	Maximum size of node
setKeys	Void	Keys	Keys of node
setMaxsize	Void	Max size	void
getChildren	LinkedList<Node>	Void	Children list
insertToParent	void	Midkey and rightnode	void
isFull	boolean	Node	True of false
findRoot	Node	Node	Root node
findLeaf	Node	Leaf Node	Leaf Node
deletePosition	double	Index and node	Double type result
contains	boolean	Index and node	True of false
findPosition	int	Index and node	The suitable position to put the key index

There are some typical functions of the Node.java class. Like:

isFull determines whether the keys in the node is less than the maximum size

findRoot will return the current node to the root node

findLeaf will return the current node to the leaf node

delete position will delete the indicated index in the current node

contains will determine whether the current node contains the index

findPosition will find the where should continue to search the index in the current node

4. IndexNode.java

this class extends the functions and variables of the Node.class. the constructor for this class should be IndexNode(int maxSize).

The main functions of the node include.

Function	Type	Input	Return
Insert	Node	key of the node	root of the tree
Split	Void	Midkey, rightnode	value of the key
Remove	Node	index	root of the tree
updateremove	Node	null	Null
Search	Key	Index	Value of results
search	ArrayList<Key>	Index range	Value list of results

For index node

For insert remove and search and range search, the index node will go to its leaf node to execute the functions.

For the updateremove function, the index node will find its siblings in this level and determine will it will merge or borrow from its siblings in this level. For more details, please check the code

5. LeafNode.java

Compared to the index node. It has two other variables, namely, leftsiblings and rightsiblings. The constructor for the leaf node should be LeafNode(int maxSize)

The main functions of the node include:

Function	Type	Input	Return
Insert	Node	key of the node	root of the tree
Split	Void	Midkey, rightnode	value of the key
Remove	Node	index	root of the tree
Search	Key	Index	Value of results
search	ArrayList<Key>	Index range	Value list of results
Getleftsibling	leafnode	Null	Leftsibling
Setleftsibling	Void	Leafnode	Lefsibling
Getrightsibling	indexnode	Null	Rightsibling
setrightsibling	void	leafnode	rightsibling

For the insert, split, remove and search range. The leaf nodes will check its siblings and determine whether it should borrow or merge with its left nodes or right nodes.

6. Key.java

This class builds up each key in the node. The variables for different type of node is also different. It has two constructors:

- 1) Key (double index, String value) for leaf nodes
- 2) Key (double index) for index nodes

Functions include

Function	Type	Input	Return
getIndex	Double	void	Index of the key
getValues	ArrayList<String>	void	Values of the key