ADS Project Report

UFID: 6110-8000
Name: Kaushik Manchenahalli Ranganatha Rao
Email: kaushik.manchena@ufl.edu

1. Introduction:
The goal of this project is to implement an in-memory B+ tree in order to use it for dictionaries. The operations supported by the implementation include,

1. Initialize (m): create a new m-way B+ tree
2. Insert (key, value)
3. Delete (key)
4. Search (key): returns the value associated with the key
5. Search (key1, key2): returns values such that in the range key1 <= key <= key2

2. Code Structure:
The code consists of the following class,

1. **bplustree.java:** This class orchestrates the entire process of performing operations on a BPlus Tree, by reading from a file and calling the appropriate methods for the above operations

Public void static main[]	
Description	This is the main method of the class

2. **BPlusTreeInstance.java:** This is the entity of the BPlus Tree which has the properties and main operations and

public void insert(*int key, int value*)	
Description	Description to handle multiple insertion scenarios which may lead to splitting of nodes to form new ones

public void searchBetweenKeys(*int start, int end*)	
Description	Method to do a range search between two keys in the leaf nodes, to find the appropriate dictionary pairs

public void delete(*int key*)	
Description	Method to delete a given key-value pair in the dictionary

3. BPlusNode.java

Super class for BPlusNode which is extended by leaf nodes and search nodes	
Keys *ArrayList<Integers>*	These are the dictionary keys

4. DataNode.java

Subclass of the BPlus nodes which maintains the Key-value pairs in the dictionary	
Values ArrayList<Double>	These Dictionary Values

public insertIntoDataNode		
Description	This method is to insert into a data node which might split the node	
Parameters	key	The key to be inserted to the DataNode
	value	The value to be inserted into the data node

5. **SearchNode.java:** The class which acts helps managing the search nodes which route the search in the BPlus Tree to get to the appropriate node

public *void* insertToSearchNode(NodeCollection obj, *int searchIndex*)		
Description	This method is responsible for inserting the entry into this node at the specified index so that it still remains sorted	
Parameters	obj	The particular entry to be inserted
	searchIndex	The index for the value

6. **NodeProcessingUtility.java:** Utility class which is responsible for handling special scenarios in the inserting and deleting from the nodes like overflows and underflows

public *void* BPlusTreeSplit(*BPlusTreeInstance, DataNode dataNode*)		
Description	Method to handle the split scenario for a data node	
Parameters	BPlusTreeInstance	Tree Instance
	DataNode	DataNode

public *void* BTreeSplit(*BPlusTreeInstance, DataNode dataNode*)		
Description	Method to handle the split scenario for a search node	
Parameters	BPlusTreeInstance	Tree Instance
	Search Node	SearchNode

public *void* processDataNodeUnderflow(*BPlusTreeInstance, DataNode leftSibling, DataNode rightSibling, SearchNode parent*)		
Description	Method to handle the split scenario for a search node	
Parameters	BPlusTreeInstance	Tree Instance
	Data Node	Left Sibling of the leaf
	Data Node	Right Sibling of the leaf
	SearchNode	Parent

public *void* processSearchNodeUnderflow(*BPlusTreeInstance, DataNode leftSibling, DataNode rightSibling, SearchNode parent*)		
Description	Method to handle the split scenario for a search node	
Parameters	BPlusTreeInstance	Tree Instance
	SearchNode	Left Sibling of the leaf
	SearchNode	Right Sibling of the leaf
	SearchNode	Parent node

7. **BPlusUtil.java : Utility class that has some common methods needed across operatons**

public static *void* isNodeFull(*BPlusNode node,BplusTreeInstance treeInstance*)		
Description	Method to handle the split scenario for a search node	
Parameters	BPlusNode	BPlusNode
	BPlusTreeInstance	TreeInstance

public static *void* isNodeDeficient(*BPlusNode node,BplusTreeInstance treeInstance*)		
Description	Method to handle the split scenario for a search node	
Parameters	BPlusNode	BPlusNode
	BPlusTreeInstance	TreeInstance