ADVANCED DATA STRUCTURES (COP 5536)

B+ TREE: PROGRAMMING PROJECT

Gayathri Manogna Isireddy

UFID: 9124 - 0699

Isireddy.g@ufl.edu

Department Of Computer & Information Science & Engineering

University of Florida - Spring 2019

Problem Description:

The Project is of creating a m-way B+ tree to store the data for efficient retrieval in a block-oriented storage context — in particular, file systems. B+ tree is a data structure used for storing key, value pairs. In this project the requirement is to develop a memory resident B+ tree (i.e., the entire tree resides in main memory). The leaves of B+ tree should be linked into a doubly linked list. The values in B+ tree is present only in leaf nodes. The main functionalities:

1. Initialize (m): Initializes a new m-way B+ tree

2. Insert (key, value): The insert function is used to insert the Key Value pair at the desired position.

3. Delete (key): Delete function deletes the key, value associated with the given key

4. Search (key): Search returns the value associated with the key

5. Range Search (key1, key2): Returns values such that the corresponding keys in the range key1 <= key <= key2

Associated Files

- BPlusTree.java: This file contains the main method, which initiates the process of reading the input file and creating the output file. On successful reading the input file makes the respective operational calls to the function in TreeOperations.java
- TreeOperations.java: This file holds all the required operations performed in developing the BPlus Tree along with Node class.
- makefile: This file helps with compilation and execution of code.

Function Prototypes and Project Flow

1. BPlusTree.java:
    - Variables used:
        - obj of type TreeOperations
          Description: An object of TreeOperations used for operation calls.
        - fileRead of type List<String>
          Description: A list to store the string lines of input file.
        - inputStream of type FileInputStream,
          Description: fileInputStream variable for given input file
        - inputBuffer of type BufferedReader
          Description: buffered reader for InputStreamReader of inputStream.
        - read of type String
          Description: string to hold readLine of inputBuffer
        - outputFile of type File,
          Description: new output file
        - writeToFile of type BufferedWriter
          Description: buffered writer used for writing the results obtained from operations performed in to the output file.
        - degree of type int
          Description: m order of the tree that need to be developed, passed as an argument for Initialize operation

- Main method flow:
  - Creates an object for TreeOperations class used in making function calls for operation present in that class.
  - Calls readInputFile method, which takes inputfile name as argument and returns the List of string read form inputfile.
  - On receiving the list of string form inputfile, checks for first string containing Initialize substring and initiates the process by making a function call to Initialize method with order value taken from string.
  - Now, looping through the other strings present in string list and making the respective function calls with the required parameters obtained from the same strings.
  - For search operations, on successful execution of the operation it gives a return value of type string, which is written in to output file using writeToFile buffer writer. If the return value from these operations is null, writes the "Null" in to the output file.
  - Finally closing the buffer writer object.
- readInputFile method:
  - This takes input file name as argument.
  - The buffered reader object created helps in reading all the line of the file and adding to a string list.

2. TreeOperations.java:
   - Variables used for Node class:
     - nodeKeys of type List<Integer>
       List of integer keys to store the keys
     - nodeValues of type List<Double>
       List of double values to store the values of keys
     - childLi of type List<Node>
       List of Nodes to store the children nodes.
     - nextEle of type Node
       Pointer to point to the next node in leaf level
     - prevEle of type Node
       Pointer to point to the previous node in leaf level
   - Default and parametrised constructors are developed

- Other Variables:
  - Root of type Node
    It acts as head of the tree
  - Order of type int
    Used to store order of the tree(m)

- Methods:
  - Initialize
    Description: Method to initialize the BPlus Tree
    Parameters: int x
    Return Type: void
  - Insert
    Description: This helps in inserting the give key, value pair to the tree.
    Parameters: int key, double value
    Return Type: void
  - insertPos
    Description: This helps in finding the position where the key need to be inserted in the node keyList.
    Parameters: Node node, int key
    Return Type: int, the position where the key can be inserted.
  - Search
    Description: This is to search for the values of given key in the tree
    Parameters: int key, key that need to be searched
    Return Type: String, value corresponding to the given key converted to string
  - Search for value in range of keys
    Description: This is to search for the values of the keys present in between the specified range of given keys.
    Parameters: int Key1, int Key2
    Return Type: String, string appended with the values of the keys present in given range.
  - Delete
    Description: This helps in deleting the key and its value from the tree.

Parameters: int key
Return Type: void

- adjucentIndex
  Description: This method is used to find the adjacent node to the given node, from where the node can borrow the key
  Parameters: Node parent, Node current_Node
  Return Type: int, index of adjacent node
- flow
  Description: This method is used in obtaining the flow that need to be followed for the given key from root to leaf level.
  Parameters: Node head, int key, List<Node> flowList
  Return Type: void

Project Flow:

- This project starts with reading the input text file and processing all the operation specified with the help of methods present in TreeOperations.java
- On each function call, the respective function gets invoked and performs the desired operations on the parameters provided.
- For both the search operation which return a string value of values corresponding to given key, which is then written in to the output file with help of buffer writer.

Execution Instruction:

The project has been tested on the tunder.cise.ufl.edu server, to test its readiness.

- On getting all the file proved on to the directory on your server, from the make file run ‘make’ command to compile the project.
- Once compiled use ‘make run’ command to run the project on the test file provided in the folder.
- This can also be executed using any input file using the below command,
  java bplusTree path_to_input_text_file

Summary

The project is successfully compiled and executed on the input file provided with the desired output.

```
121.56
3.55,-3.95
-3.95

```