HACETTEPE UNIVERSITY COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

Name-Surname : Nehil DANIŞ

Student ID : 21327876

Advisers : R.A. Ali ÇAĞLAYAN , Dr. Adnan ÖZSOY , Dr. Erkut ERDEM , Dr. Gönenç ERCAN

Subject : Data Organization with B-tree Structure

Environment : JDK8

Programming Language : JAVA

1.Problem

Here in this experiment we were expected to implement an index system by using B-Tree structure . There could be only N numbers in one node .If you wanted to add another item to the node , when the key number of node is equal to N, the node will be divided into two. The N/2 . key will go to the parent node. The keys from 0 to N/2-1 will go to the one child. The keys from N/2+1 to N will go to the another child. When one key was added to the one node the keys of node sorted in ascending order . If a word had already been added to the B-Tree , we are expected to find this word in B-Tree and add a new file element for new added word to the finding key . B-Tree does not allow adding identical elements .

2.Solution

I divide problem into parts to implement this experiment . At first I read the files and then I implemented B-Tree by using numbers . And I learned how B-Tree behaves when we add new element or adding the (n+1). element to the one node . After understanding what B-Tree is , then I unified the read part and B-Tree implementation part . Then I tried lots of different input . By doing this I handled lots of different error . Here in the experiment we are expected for searching individual element , searching elements that start with specific string , finding files of the two specific elements when they involved in the same file , finding the files names that has just first one of the specific elements , not the second one , and level order and in order traverse . After implementing B-Tree I added the codes of these functions to the project .

3.Algorithm

1. Reading path and finding the readable files.

1.1. Read each line and split each of them depend on specific regex .

1.1.1. Deciding the index and row of each word.

1.1.2. Send each element to the insert function for adding the B-tree

1.1.2.1. Control the existence of word.

1.1.2.1.1. If it has already added , the program will just add the file name and index of this new word the the existence key.

1.1.2.1.2. If it does not exist , we we add new key to the consistent node.

1.1.2.1.2.1. If the size of the node that we added a new key , is more than or equal to n+1 the node will be divided into two new nodes. The middle element goes to the parent of divided element . The children nodes of the parent node of divided element will find their own place .

2.Reading commands file.

2.1. Split each line of commands file by using space delimiter . And then decide the which function will be implemented.

2.1.1. If the first element of divided command line is traverse , there are two function that will be implemented.

2.1.1.1. If the second element of divided line is in-order , we will printed B-Tree in in order sequence.

2.1.1.2. If the second element of divided line is level-order , we will printed B-Tree level order sequence.

2.1.2. If the first element of divided command line is search , there are four function that will be implemented.

2.1.2.1. Searching individual element.

2.1.2.2. Searching elements that start with specific string .

2.1.2.3. Searching the files that two specific elements involved in .

2.1.2.4. Searching the files that first element of specific elements is involved in and the second is not .

4.Usage of this algorithm

This algorithm help us to find the places of elements in lots of different files and reach these elements quickly by using B-Tree . If you add the path way of files that you wanted to read , the stop words files and also the command and output files to the system , the system will be implemented each command that you asked .

5.Notes

I came up with lots of different problems when I tried to implement this experiment. One of the hardest one is about regex. Because of this problem I thinked that I had an error in the B-Tree implementation . But the cause of problem is trying to add an element that is length is 0 . This caused a problem in level order traverse ,then I implemented in right way ,the code was executed clearly.