Lab3
BTree Implementation
&
Search Engine

Name:

Makrm William

ID:

64

1)Code Design “SearchEngine”:

- First we made a help Class named “ResultSearch” which contain of a list of “ISearchResult” and doing the add & delete from list.
- The class “SearchResult” is a simply class contains of setters and getters for ID & Rank.
- The class SearchEngine has only one attribute which a BTree of String & ResultSearch and the following functions:

1. indexWebPage:

it takes a one parameter String path and check if not null and exists, then call the “getdoc” function which return a document to use DOM XML Parser, by parsing we get the ID & “Textcontent” of the file, then we split the textcontent to “words” to insert it in the BTree using the class “ResultSearch” to put ID & Rank.

2. indexDirectory:

it also take one parameter String path, also check it not null and exists, then call a recursion function “getnames” which check all path if File put its path and if directory call itself. Then call “ indexWebPage” for each Path.

3. deleteWebpage:

   take same parmter and call same function to get document and DOM XML Parsing file as first function but it uses delete function in BTree and ResultSearch

4. SearchByWordWithRanking:

   It also take one parameter String word and check it not null and exists, then call BTree Search by “word”
   It return a object of “ResultSearch” classs , then using “gettArr” to get the List of ISearchResult and sort them by Rank as in “UnitTest” and return it .

5. SearchByMultipleWordByRanking:

   it take one sentence String and check it not null, then split it to words , check arr empty return empty List else Search for first word and put it in List, then for each other word in sentence Search for it and compare the ID to put Min Rank and add Result to original List, then also sort it as” UnitTest” and return it.

1) Time and space:

   1) BTree

      Assume height of tree h , maximum degree = m, d=m/2.

      Then h < $\log_d(n+1)/2 + 1$

      - GetMinmumDegree:

        Takes O(1), with only int space.

      - GetRoot:

        Takes O(1), with only INode space.

      - Insert:

        Takes at most 3h access and comparison so it takes O(h) with array list for keys&values&children.

      - Search:

        Takes at most h comparison so it takes O(h),with only pointer INode.

2)SearchEngine:

Assume number of files in directory= v, number of doc in file =n, number of words in doc file=m.

- indexWebPage:
  it takes O(m*n*h), with array of m and Arraylist of ID &Rank

- indexDirectory:
  it takes O(v*m*n*h), with Arraylist of v and array of m and Arraylist of ID & Rank.

- deleteWebpage:

it takes O(m*n*h), with array of m and Arraylist of ID &Rank

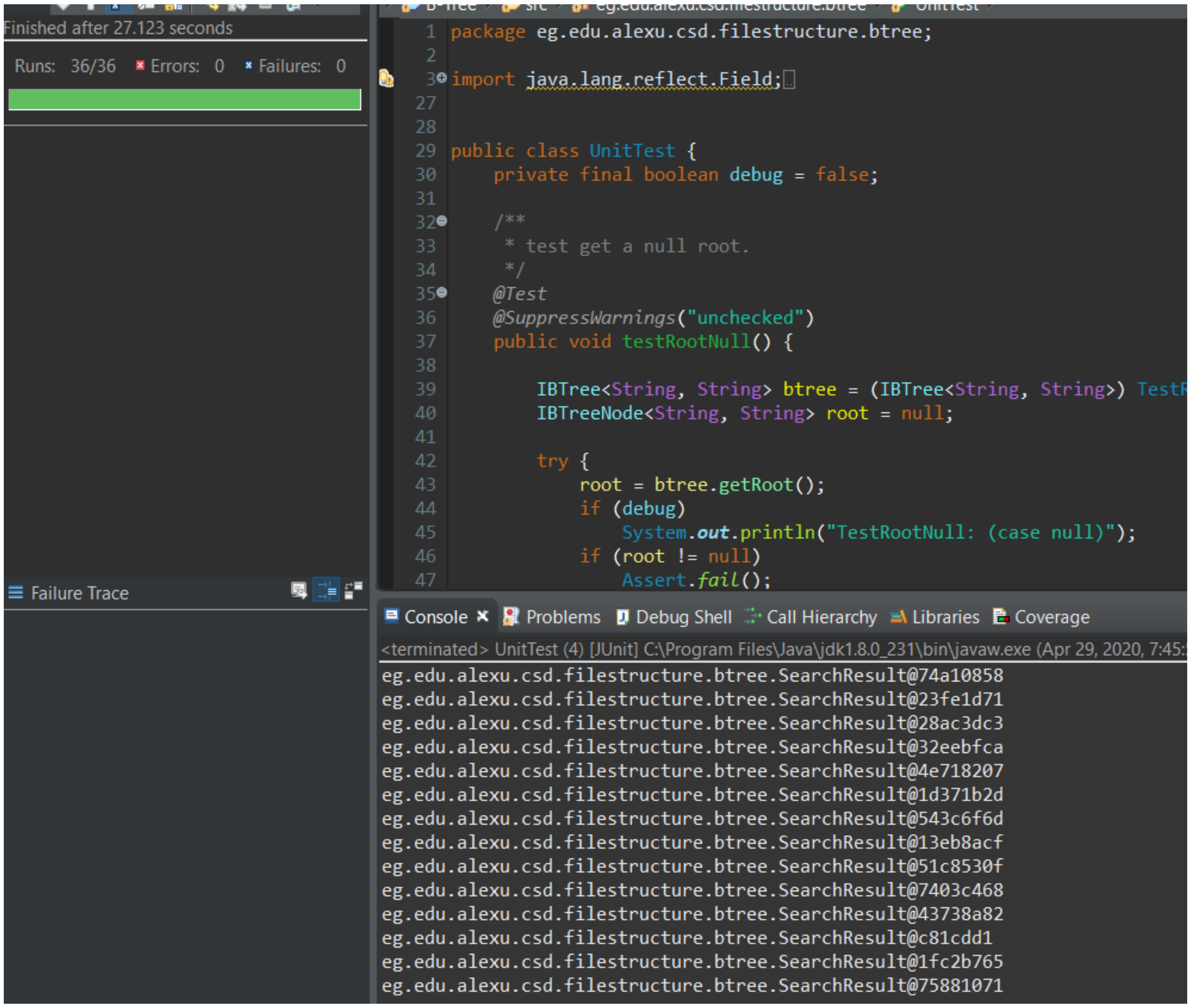
- SearchByWordWithRanking:
  It takes O(h) for search in BTree and O(logn) for sort
  So it takes O(h+logn), with only Arraylist of result

- SearchByMultipleWordByRanking:

  It takes O(h) for search for each word and O(N^3) for comparing all ID for each word and O(logn) for sort.

  So it takes O(h+N^3+logn).

3)UnitTest :



Finished after 27.123 seconds

Runs: 36/36 Errors: 0 Failures: 0

Failure Trace

```java
package eg.edu.alexu.csd.filestructure.btree;

import java.lang.reflect.Field;


public class UnitTest {
    private final boolean debug = false;

    /**
     * test get a null root.
     */
    @Test
    @SuppressWarnings("unchecked")
    public void testRootNull() {

        IBTree<String, String> btree = (IBTree<String, String>) TestR
        IBTreeNode<String, String> root = null;

        try {
            root = btree.getRoot();
            if (debug)
                System.out.println("TestRootNull: (case null)");
            if (root != null)
                Assert.fail();
```

Console Problems Debug Shell Call Hierarchy Libraries Coverage

```
<terminated> UnitTest (4) [JUnit] C:\Program Files\Java\jdk1.8.0_231\bin\javaw.exe (Apr 29, 2020, 7:45:
eg.edu.alexu.csd.filestructure.btree.SearchResult@74a10858
eg.edu.alexu.csd.filestructure.btree.SearchResult@23fe1d71
eg.edu.alexu.csd.filestructure.btree.SearchResult@28ac3dc3
eg.edu.alexu.csd.filestructure.btree.SearchResult@32eebfca
eg.edu.alexu.csd.filestructure.btree.SearchResult@4e718207
eg.edu.alexu.csd.filestructure.btree.SearchResult@1d371b2d
eg.edu.alexu.csd.filestructure.btree.SearchResult@543c6f6d
eg.edu.alexu.csd.filestructure.btree.SearchResult@13eb8acf
eg.edu.alexu.csd.filestructure.btree.SearchResult@51c8530f
eg.edu.alexu.csd.filestructure.btree.SearchResult@7403c468
eg.edu.alexu.csd.filestructure.btree.SearchResult@43738a82
eg.edu.alexu.csd.filestructure.btree.SearchResult@c81cdd1
eg.edu.alexu.csd.filestructure.btree.SearchResult@1fc2b765
eg.edu.alexu.csd.filestructure.btree.SearchResult@75881071
```