GTU DEPARTMENT of COMPUTER ENGINEERING

CSE222/505 – SPRING 2023

HOMEWORK 5 REPORT

ÖMER FARUK ÇOLAKEL

200104004043

1. SYSTEM REQUIREMENTS

```
private static void ReadTxtFile(String filename, DefaultMutableTreeNode root)
```

This method requires the name of the file that we need to read from and root of the tree. Tree node is not being used in this method but I used it while calling addToJtree.

```
public static void addToJtree(DefaultMutableTreeNode root, DynamicArray data)
```

This method requires root to add nodes and a dynamic array (2D string array) that holds the name and location of the nodes.

```
public static void BFSAlgorithm(DefaultMutableTreeNode root)
```

This method requires the root of the tree. It uses scanner to read the terminal. It gets the node's name that will be searched.

```
public static void DFSAlgorithm(DefaultMutableTreeNode root)
```

This method requires the root of the tree. It uses scanner to read the terminal. It gets the node's name that will be searched.

```
public static void PostOrderTraversalAlgorithm(DefaultMutableTreeNode root)
```

This method requires the root and uses scanner to get the node that will be searched from terminal.

```
public static int PostOrderTraversalAlgorithmHelper(DefaultMutableTreeNode root, String str, int count)
```

This method requires root of the tree (or the subtree), the node that it is searching and number of steps.

```
public static void moveFromJTree(DefaultMutableTreeNode root)
```

This method requires the root of the tree. It uses scanner to get source and destination from terminal.

Dynamic Array

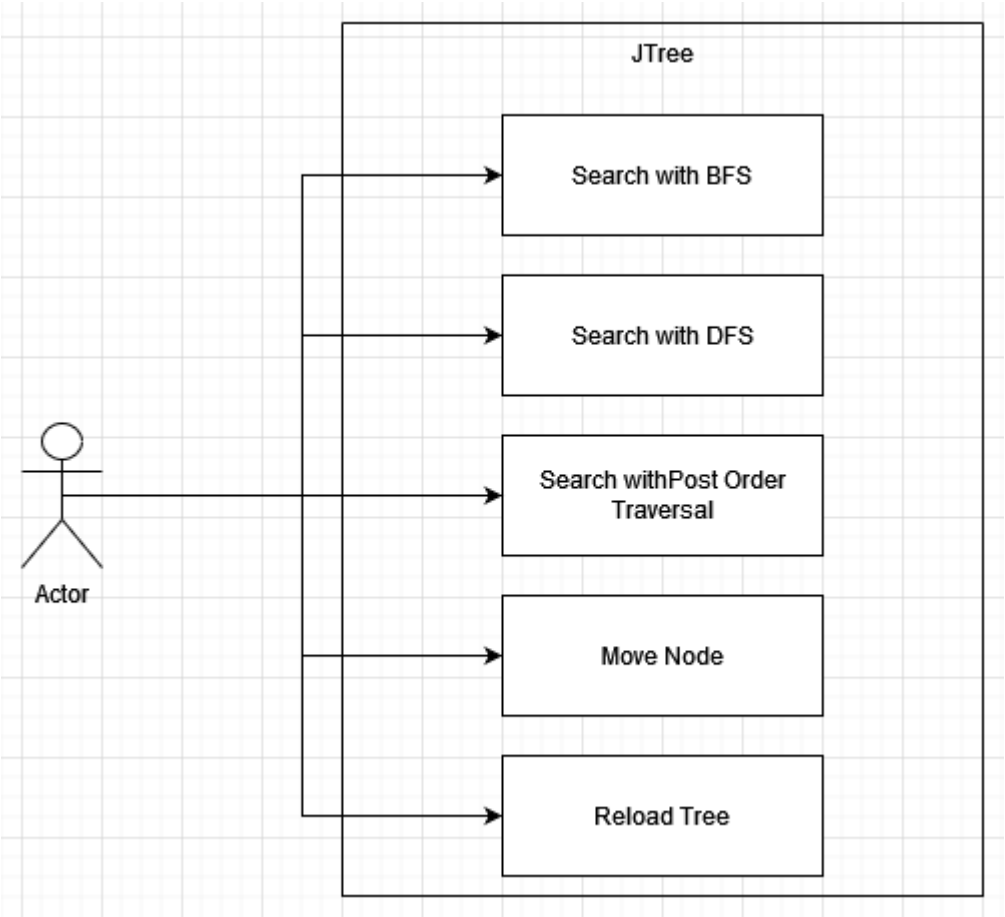
```
public void add(String[] strArr)
```

This method requires a 1D array to add to dynamic array.

```
public String get(int i, int j)
```

This method requires row and column index to return the string in the dynamic array.

2. USE CASE DIAGRAM



3. PROBLEM SOLUTION APPROACH

While creating the tree, I analysed every row one by one. If one of the nodes were already present, I didn't add it to the tree and just navigated to it.

While searching with BFS algorithm, I added the root to a queue and then added the children of it. Then those children added their children also. With this, we search the tree top to bottom.

While searching with DFS algorithm, I pushed the root to a stack and it pushed its children to stack. So last added to stack and its children checked first.

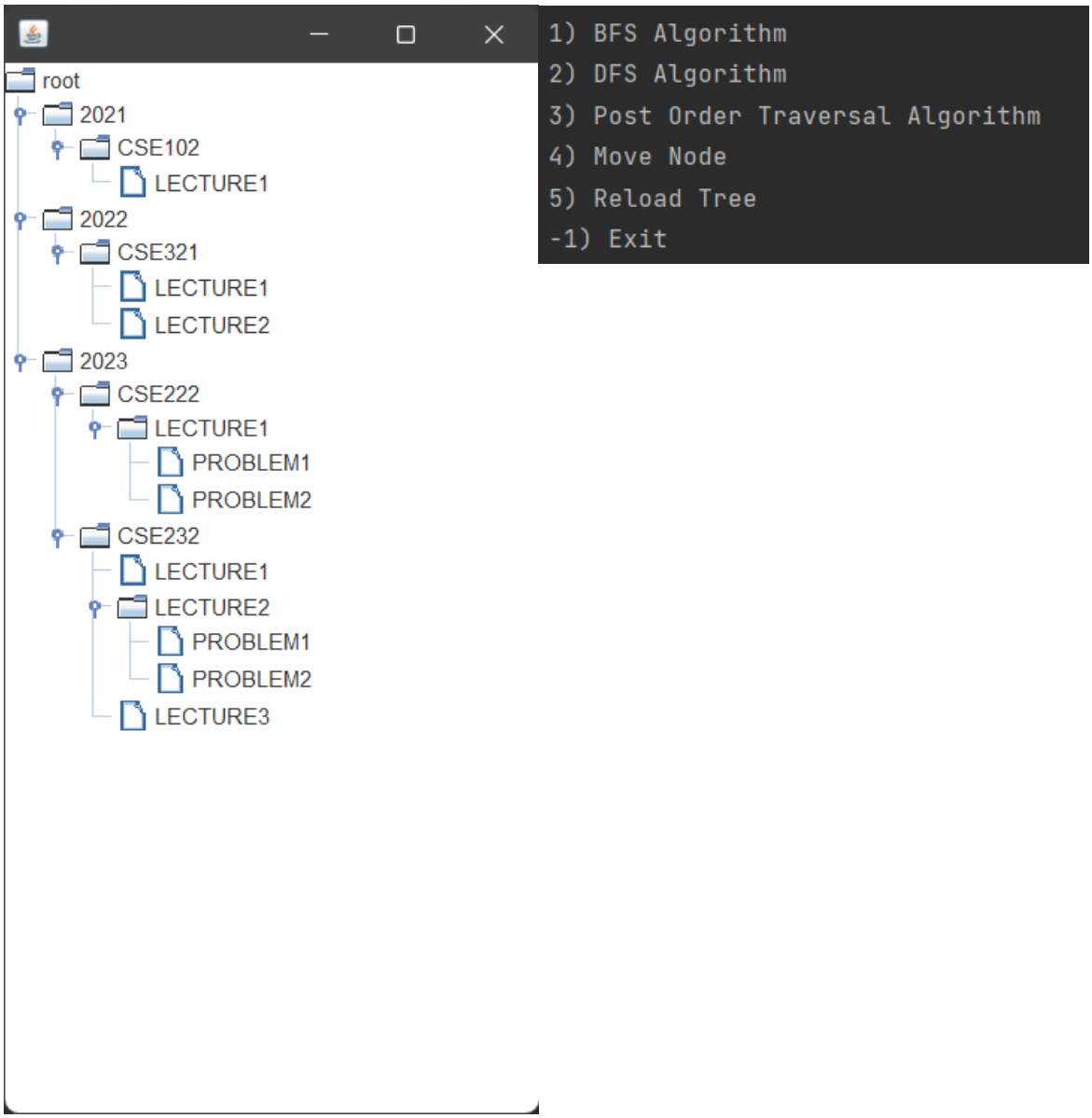
While searching with post order traversal algorithm, we check the left of the node then right of the node and lastly the node itself. So, it checks the children starting from 0 to end of the children array and then the parent node. Helper method is recursive and it traverses through all children.

While moving a node, it checks if the source is valid first. Then goes to the destination (creates nodes if needed) and then continues according to source. It checks if the source node already exists, if it's then overwriting it.

While reloading the tree, it just reads the same file again and deletes the old tree.

4. TEST CASES and RESULTS

Running the Program with Valid Text File



Running with Invalid Text File

```
An error occurred.
java.io.FileNotFoundException Create breakpoint :
    at java.base/java.io.FileInputStream.ope
    at java.base/java.io.FileInputStream.ope
    at java.base/java.io.FileInputStream.<in
    at java.base/java.util.Scanner.<init>(Sc
    at Main.ReadTxtFile(Main.java:89)
    at Main.main(Main.java:15)
```

Searching CSE232 with BFS Algorithm

```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
1
Enter the node you want to search for
CSE232
Using BFS Algorithm to find CSE232 in the tree...
Step 1 -> root
Step 2 -> 2021
Step 3 -> 2022
Step 4 -> 2023
Step 5 -> CSE102
Step 6 -> CSE321
Step 7 -> CSE222
Step 8 -> CSE232(Found!)
```

Searching CSE2332 with BFS Algorithm

```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
1
Enter the node you want to search for
CSE2332
Using BFS Algorithm to find CSE2332 in the tree...
Step 1 -> root
Step 2 -> 2021
Step 3 -> 2022
Step 4 -> 2023
Step 5 -> CSE102
Step 6 -> CSE321
Step 7 -> CSE222
Step 8 -> CSE232
Step 9 -> LECTURE1
Step 10 -> LECTURE1
Step 11 -> LECTURE2
Step 12 -> LECTURE1
Step 13 -> LECTURE1
Step 14 -> LECTURE2
Step 15 -> LECTURE3
Step 16 -> PROBLEM1
Step 17 -> PROBLEM2
Step 18 -> PROBLEM1
Step 19 -> PROBLEM2
CSE2332 Not Found!
```

Searching CSE232 with DFS Algorithm

```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
2
Enter the node you want to search for
CSE232
Using DFS Algorithm to find CSE232 in the tree...
Step 1 -> root
Step 2 -> 2023
Step 3 -> CSE232(Found!)
```

Searching CSE2332 with DFS Algorithm

```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
2
Enter the node you want to search for
CSE2332
Using DFS Algorithm to find CSE2332 in the tree...
Step 1 -> root
Step 2 -> 2023
Step 3 -> CSE232
Step 4 -> LECTURE3
Step 5 -> LECTURE2
Step 6 -> PROBLEM2
Step 7 -> PROBLEM1
Step 8 -> LECTURE1
Step 9 -> CSE222
Step 10 -> LECTURE1
Step 11 -> PROBLEM2
Step 12 -> PROBLEM1
Step 13 -> 2022
Step 14 -> CSE321
Step 15 -> LECTURE2
Step 16 -> LECTURE1
Step 17 -> 2021
Step 18 -> CSE102
Step 19 -> LECTURE1
CSE2332 Not Found!
```

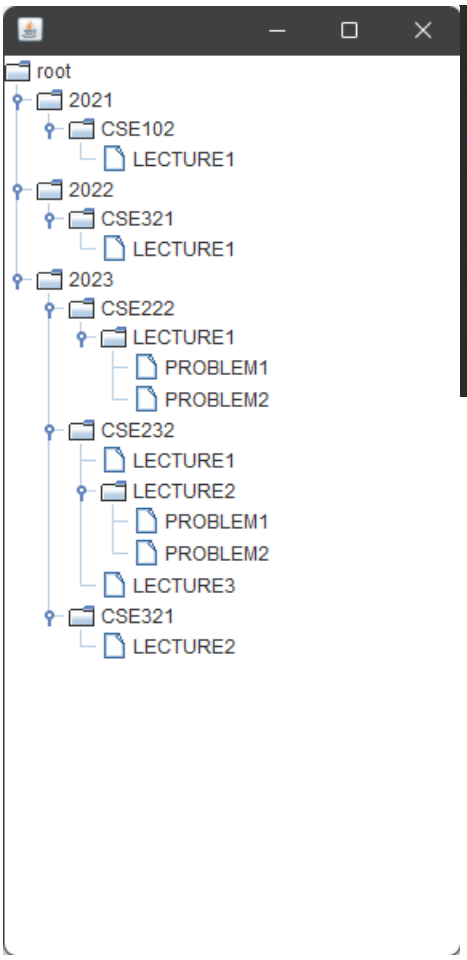
Searching CSE232 with Post Order Traversal Algorithm

```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
3
Enter the node you want to search for
CSE232
Using Post Order Traversal Algorithm to find CSE232 in the tree...
Step 1 -> LECTURE1
Step 2 -> CSE102
Step 3 -> 2021
Step 4 -> LECTURE1
Step 5 -> LECTURE2
Step 6 -> CSE321
Step 7 -> 2022
Step 8 -> PROBLEM1
Step 9 -> PROBLEM2
Step 10 -> LECTURE1
Step 11 -> CSE222
Step 12 -> LECTURE1
Step 13 -> PROBLEM1
Step 14 -> PROBLEM2
Step 15 -> LECTURE2
Step 16 -> LECTURE3
Step 17 -> CSE232(Found!)
```

Searching CSE2332 with DFS Algorithm

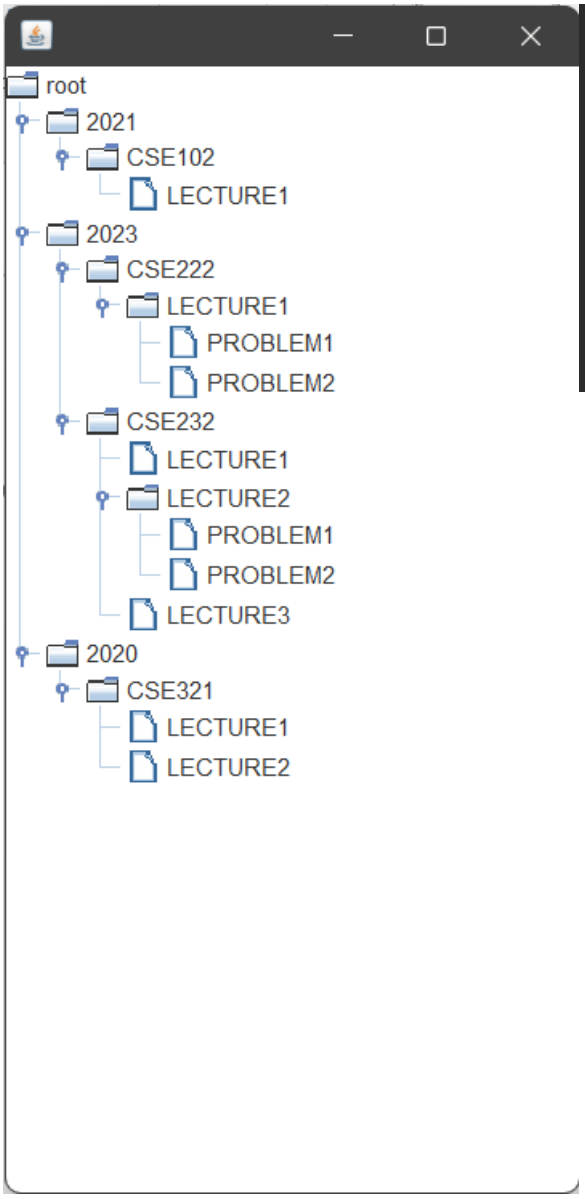
```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
3
Enter the node you want to search for
CSE2332
Using Post Order Traversal Algorithm to find CSE2332 in the tree...
Step 1 -> LECTURE1
Step 2 -> CSE102
Step 3 -> 2021
Step 4 -> LECTURE1
Step 5 -> LECTURE2
Step 6 -> CSE321
Step 7 -> 2022
Step 8 -> PROBLEM1
Step 9 -> PROBLEM2
Step 10 -> LECTURE1
Step 11 -> CSE222
Step 12 -> LECTURE1
Step 13 -> PROBLEM1
Step 14 -> PROBLEM2
Step 15 -> LECTURE2
Step 16 -> LECTURE3
Step 17 -> CSE232
Step 18 -> 2023
Not Found!
```

Moving 2022->CSE321->LECTURE2 to 2023




```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
4
Enter the node you want to move
2022->CSE321->LECTURE2
Enter the node you want to move to
2023
```

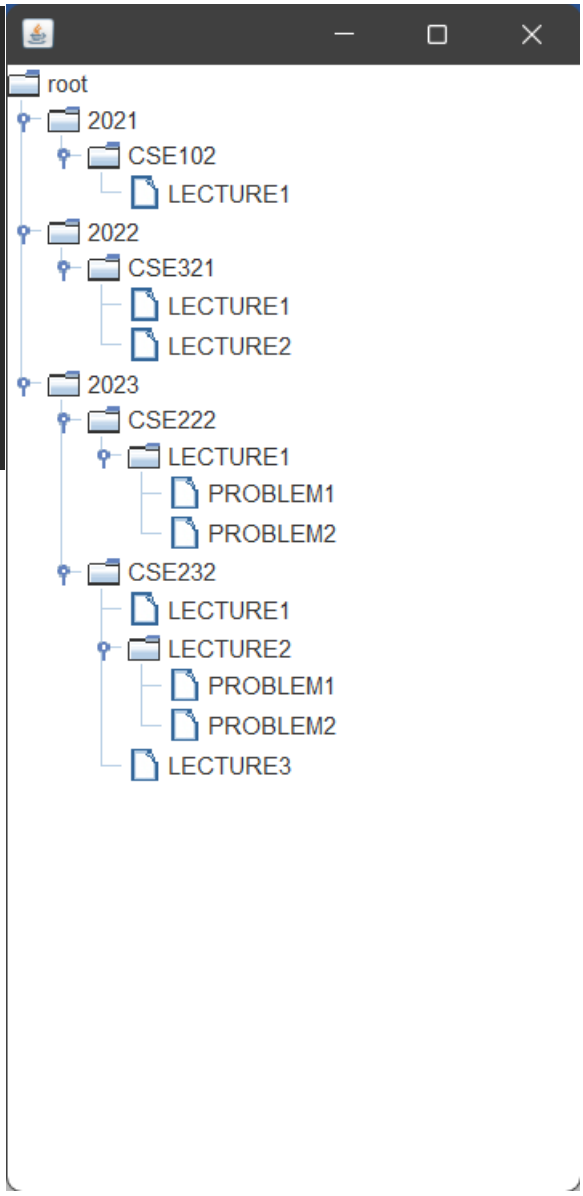
Moved 2022->CSE321to 2020



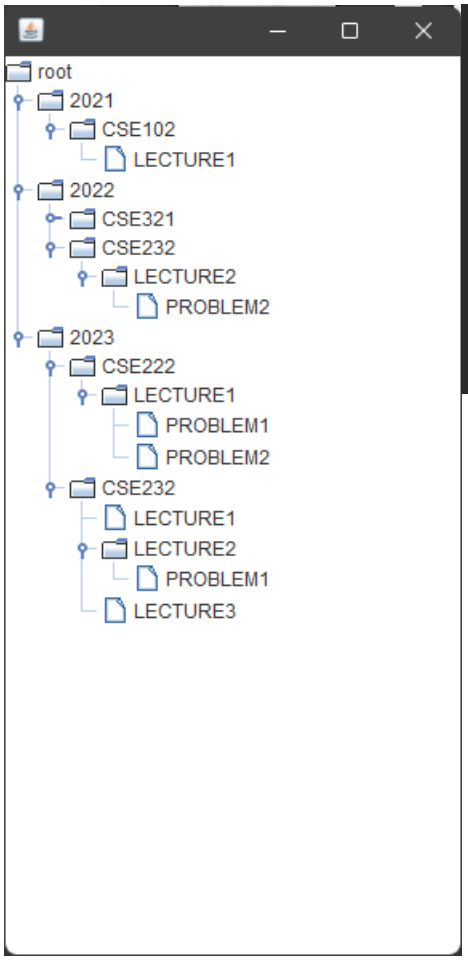

```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
4
Enter the node you want to move
2022->CSE321
Enter the node you want to move to
2020
```

Moved 2022->CSE222 to 2020

```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
4
Enter the node you want to move
2022->CSE222
Enter the node you want to move to
2020
Can't move 2022->CSE222
Because it doesn't exist in the tree
```

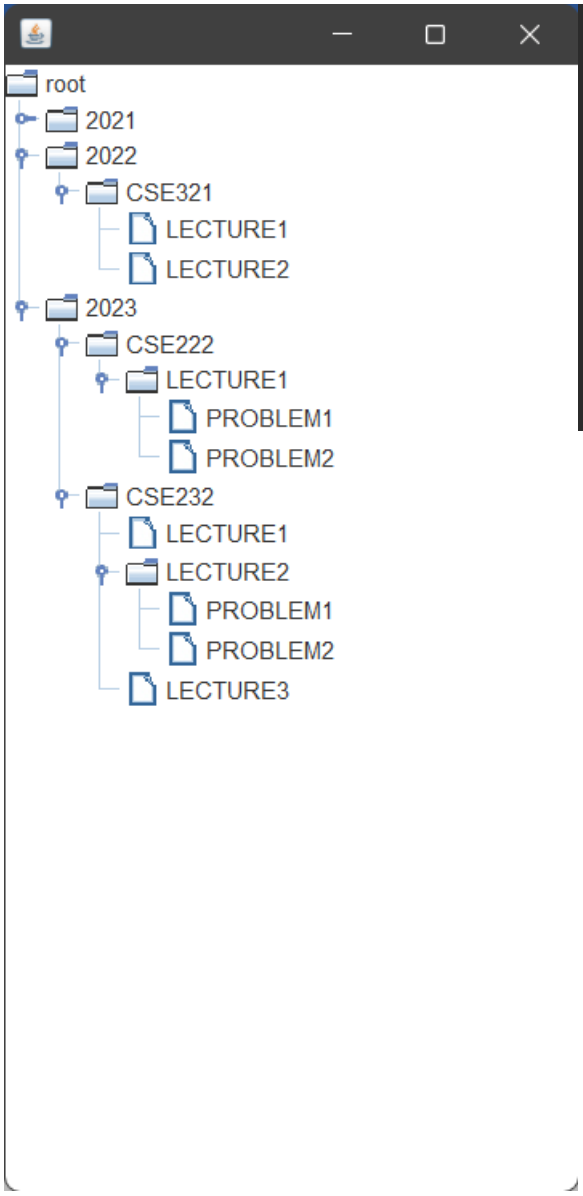


Moved 2023->CSE232->LECTURE2->PROBLEM2 to 2022




```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
4
Enter the node you want to move
2023->CSE232->LECTURE2->PROBLEM2
Enter the node you want to move to
2022
```

Moved 2023->CSE321->LECTURE1to 2022




```
1) BFS Algorithm
2) DFS Algorithm
3) Post Order Traversal Algorithm
4) Move Node
5) Reload Tree
-1) Exit
4
Enter the node you want to move
2023->CSE321->LECTURE1
Enter the node you want to move to
2022
2023->CSE321->LECTURE1 has been overwirtten!
```

5. CLASS DIAGRAM