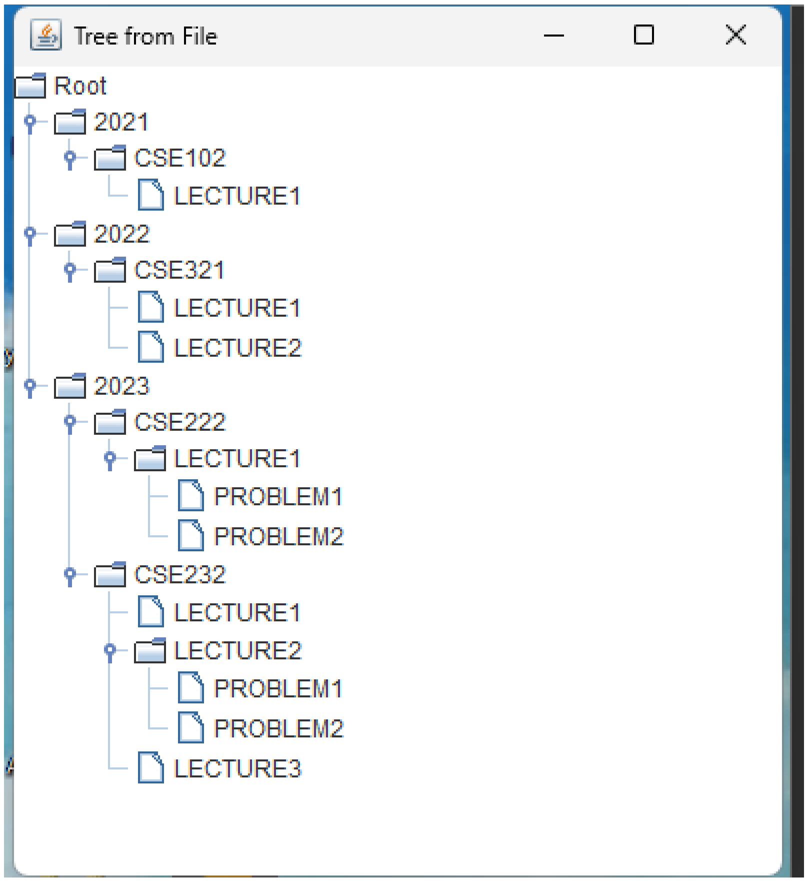
CSE222 HW5

.

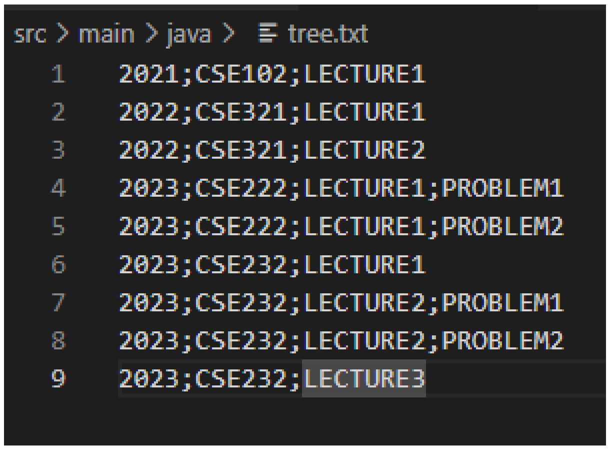
MAY 4, 2023

ELIFNUR KABALCI
1801042617

Part A: In this part, the values read from the file were turned into a tree and printed to the screen via jframe. With Jtree, nodes could be viewed one-to-one.



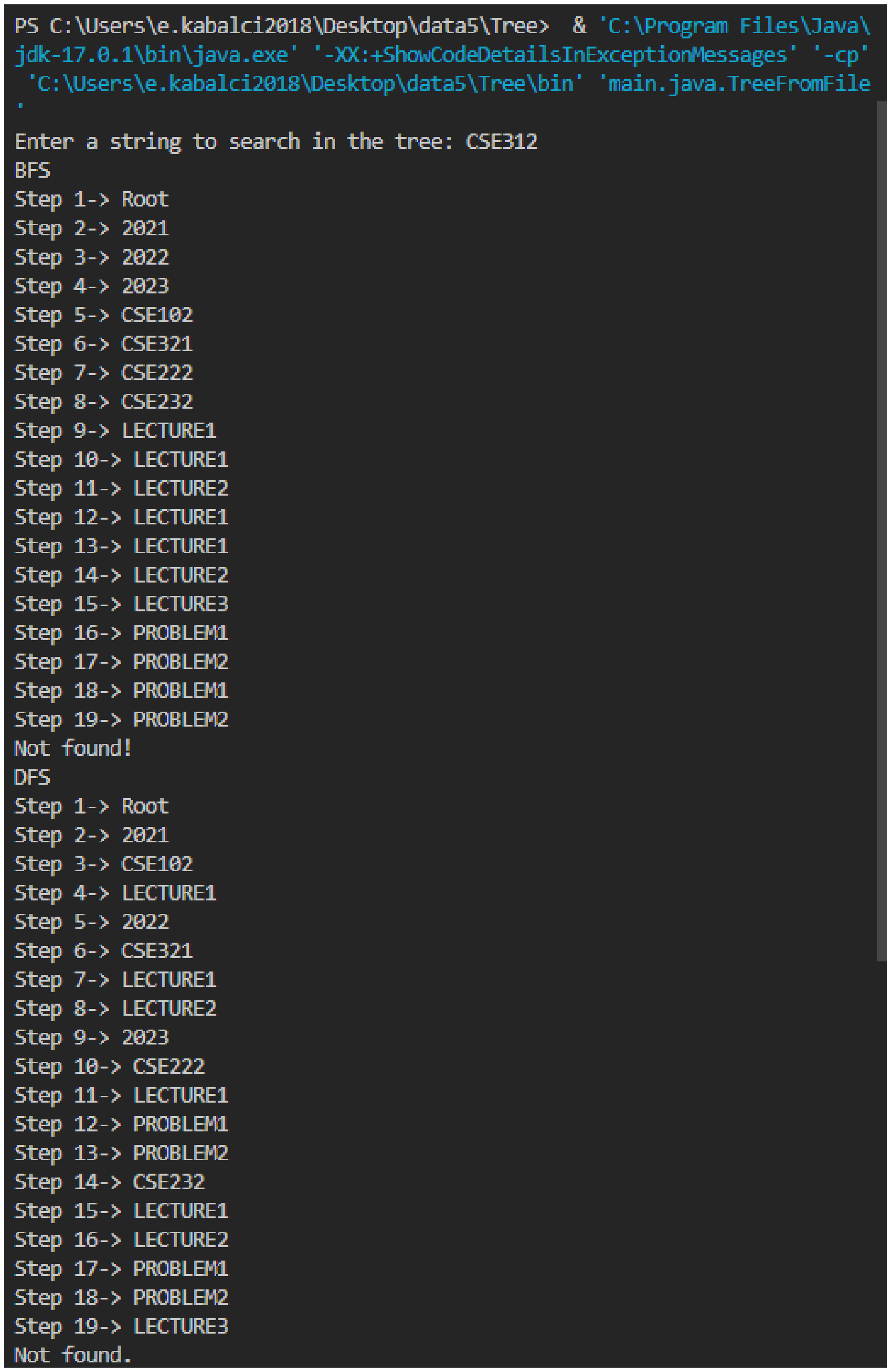
```
Tree from File
Root
  2021
    CSE102
      LECTURE1
  2022
    CSE321
      LECTURE1
      LECTURE2
  2023
    CSE222
      LECTURE1
        PROBLEM1
        PROBLEM2
    CSE232
      LECTURE1
      LECTURE2
        PROBLEM1
        PROBLEM2
      LECTURE3
```



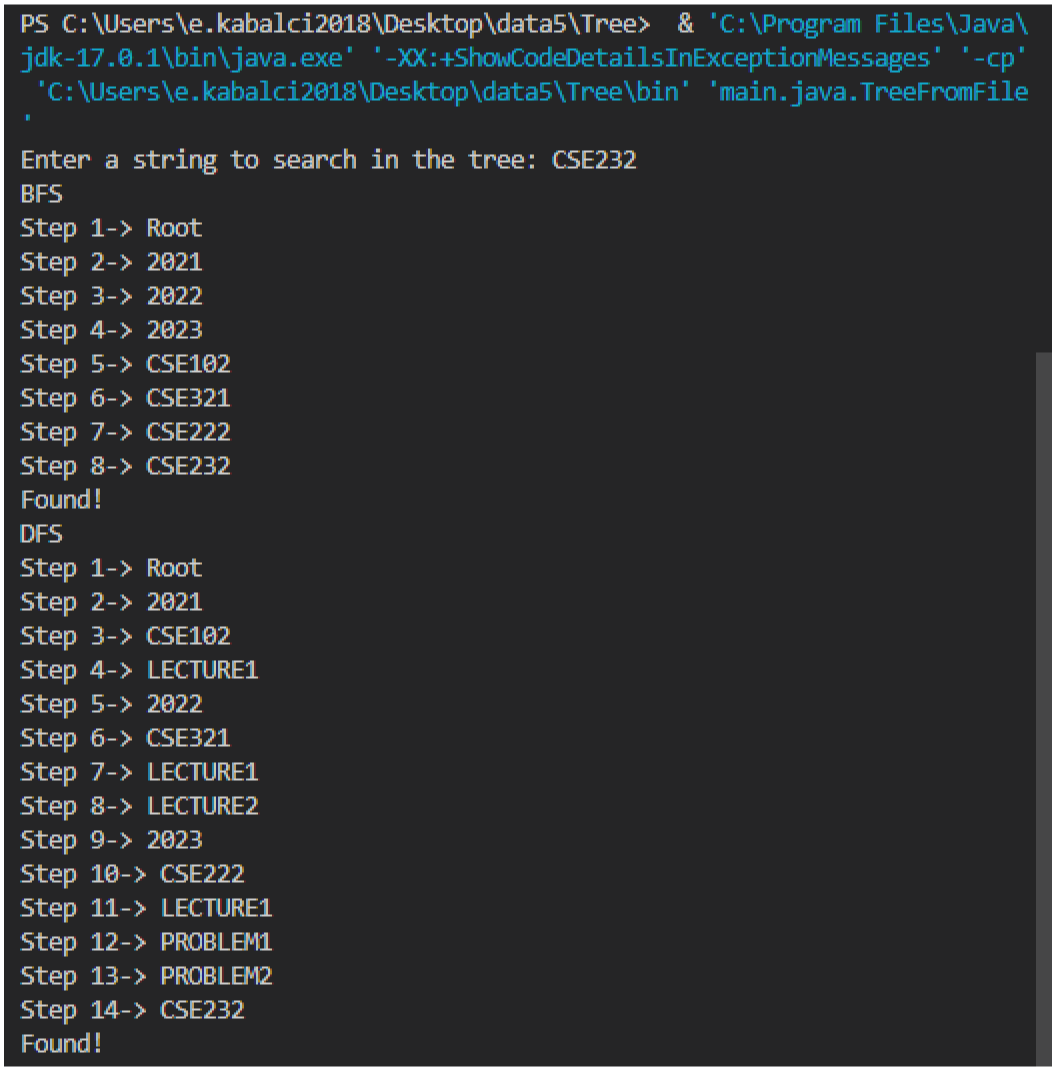
src > main > java > tree.txt

```
2021;CSE102;LECTURE1
2022;CSE321;LECTURE1
2022;CSE321;LECTURE2
2023;CSE222;LECTURE1;PROBLEM1
2023;CSE222;LECTURE1;PROBLEM2
2023;CSE232;LECTURE1
2023;CSE232;LECTURE2;PROBLEM1
2023;CSE232;LECTURE2;PROBLEM2
2023;CSE232;LECTURE3
```

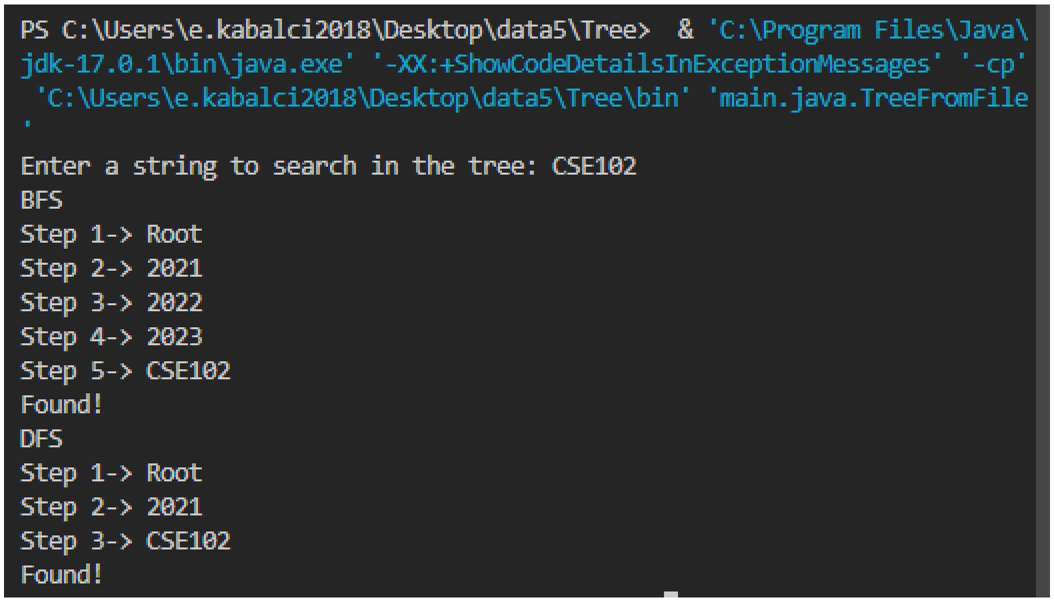
Part B – Part C: Part B and Part C are also searched with BFS and DFS. I took a screenshot of the two together to show their differences. Since the CSE312 lesson is not included in the tree, the result of searching all of them is suppressed. Others terminate the program belonging to their part when they find it.



```
PS C:\Users\e.kabalci2018\Desktop\data5\Tree>  & 'C:\Program Files\Java\jdk-17.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\e.kabalci2018\Desktop\data5\Tree\bin' 'main.java.TreeFromFile'
Enter a string to search in the tree: CSE312
BFS
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
Not found!
DFS
Step 1-> Root
Step 2-> 2021
Step 3-> CSE102
Step 4-> LECTURE1
Step 5-> 2022
Step 6-> CSE321
Step 7-> LECTURE1
Step 8-> LECTURE2
Step 9-> 2023
Step 10-> CSE222
Step 11-> LECTURE1
Step 12-> PROBLEM1
Step 13-> PROBLEM2
Step 14-> CSE232
Step 15-> LECTURE1
Step 16-> LECTURE2
Step 17-> PROBLEM1
Step 18-> PROBLEM2
Step 19-> LECTURE3
Not found.
```



```
PS C:\Users\e.kabalci2018\Desktop\data5\Tree>  & 'C:\Program Files\Java\
jdk-17.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
 'C:\Users\e.kabalci2018\Desktop\data5\Tree\bin' 'main.java.TreeFromFile
'
Enter a string to search in the tree: CSE232
BFS
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!
DFS
Step 1-> Root
Step 2-> 2021
Step 3-> CSE102
Step 4-> LECTURE1
Step 5-> 2022
Step 6-> CSE321
Step 7-> LECTURE1
Step 8-> LECTURE2
Step 9-> 2023
Step 10-> CSE222
Step 11-> LECTURE1
Step 12-> PROBLEM1
Step 13-> PROBLEM2
Step 14-> CSE232
Found!
```



```
PS C:\Users\e.kabalci2018\Desktop\data5\Tree>  & 'C:\Program Files\Java\
jdk-17.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
 'C:\Users\e.kabalci2018\Desktop\data5\Tree\bin' 'main.java.TreeFromFile
'
Enter a string to search in the tree: CSE102
BFS
Step 1-> Root
Step 2-> 2021
Step 3-> 2022
Step 4-> 2023
Step 5-> CSE102
Found!
DFS
Step 1-> Root
Step 2-> 2021
Step 3-> CSE102
Found!
```

How can I design:

Part A:

I first created a treesystem object. I read the txt file with this object. I did this reading with the readfromfile method. In this method, I used Scanner first, but it gave an error. To solve this error, I replaced the read element with buffer and put IOexception throw in the file definition. I read the file line by line and kept the data I read in a one-dimensional arraylist. Then I combined these one dimensional lists into a 2 dimensional list. Since I did this privately inside the general class structure, I was able to access directly from other files. Then I used the createtree method to convert the data I received into a tree structure. In this method, I processed the data on the root variable, which I previously defined as private, and turned it into a tree. In the design here, I used a for loop that goes up to the row each time in the for loop as much as the number of rows of the data list we read from the file and record. This generated approximately O(n^2) complexity. I have a for loop that returns as much as getchild inside these two for loops. So my complexity is O(n^3). If the value of each column in the rows of getdata is the same as the name of the generated node, I write it there and thus the tree is created. Then the show frame method works. Here I've done suppression using jtree and tframe. I adjusted the size and visibility of the frame. It's also set to turn off in case of pressing the cross.

Part B:

I used queue as auxiliary storage in this part. I keep count for the number of steps to print the screen. I added the tree structure kept as GetRoot to the queue. In the established while loop, I deleted an element from the queue in each round. So I was able to check while I isempty. If the node running on it is not zero, it means we will continue to work, so I suppressed the number of steps and increased the count. If the string of the node is the same as the input, it says found. I have added the children of the given node back to queue if not found yet. If it has exited while and hasn't found it yet, it says I couldn't find it.

Part C:

I used stack as a helper in this part. I kept count to print the step screen again. The general struct structure of df is the same as bf. I do the general check with while. I continue by deleting elements from the stack at each turn and adding their children back to the stack just before the loop ends. When it finds the data, I print it found on the screen and close the code. Here complexity is generally O(n) but. It rises to O(n^2) with the for loop used at the bottom.

General:

I do my general control in main. I received the data to be searched from the user once and tried all of them. I wrote the PostOrderTraversal function and it was finding the data there but not closing the program. And I couldn't print the steps correctly. So I removed it from the code.

I also wrote the Move function, but it was also directly deleting the deleted directory there. That's why I removed it from the code as well.