**CSE 464 Readme.md**

**GitHub Repo Link: https://github.com/divyavijayakumar72/CSE_464_2023**

CI → https://github.com/divyavijayakumar72/CSE_464_2023/actions

BFS -> https://github.com/divyavijayakumar72/CSE_464_2023/tree/bfs/src/main/java/edu.asu.scai.rise

DFS -> https://github.com/divyavijayakumar72/CSE_464_2023/tree/dfs

**INTRODUCTION:**

The goal of this project is to convert a DOT file to a graph object and perform graph manipulations on it. I have created a custom graph class, **GraphManager.java** where I have used a Hashmap to represent the graph.

I used **Graphviz-java api library** to convert DOT file to a graph object. I have used **Main.java** class to run the project.

**FEATURE 1:**

The Main class contains the following code for creating an instance of the GraphManger class and to execute Feature 1 features.

Running the below code in the Main method will execute the GraphManager class for the Feature 1 implementations. I have created an input file **input.dot** that contains a digraph with 4 nodes and 4 edges. For generating an output file, I have specified the file name as **output.txt**. Running outputGraph() method will generate a text file called output.txt in the project folder.

```
GraphManager<String> graph = new GraphManager<>();

graph.parseGraph("input.dot");

graph.countNodes();

graph.getLabel();

graph.countEdges();

graph.getEdgeDirection();

graph.containsEdge("a", "b");

graph.toString();

graph.outputGraph("output.txt");
```

**FEATURE 2:**

Running the below lines of code in the Main class will execute the Feature 2 code implementation like adding and removing node and list of nodes. I have given input for the node list to be added and removed as follows:

```
String nodeList[] = {"q", "w", "r", "t", "h", "m"};
String nodeListRemoved[] = {"w", "r"};
graph.addNode("e");
graph.addNodes(nodeList);
graph.removeNode("e");
graph.removeNodes(nodeListRemoved);
```

**FEATURE 3:**

Running the below lines of code in the Main class will execute the Feature 3 code implementation like adding and removing edges from the graph. I have given input for the edges to be added and removed as follows:

```
graph.addEdge("q", "t");
graph.removeEdge("q", "t");
```

**FEATURE 4:**

Running the below lines of code in the Main class will execute the Feature 4 code implementation like converting the graph object to DOT file and then to PNG format.

```
graph.outputDOTGraph("output.dot");
graph.outputGraphics("response.png", "png");
```

**CONCLUSION:**

Therefore, **running the Main.java class** will call the methods implemented in GraphManager.java class will execute the features specified in the assignment instructions. The **GraphManagerTest.java** class contains the unit test cases for each feature.

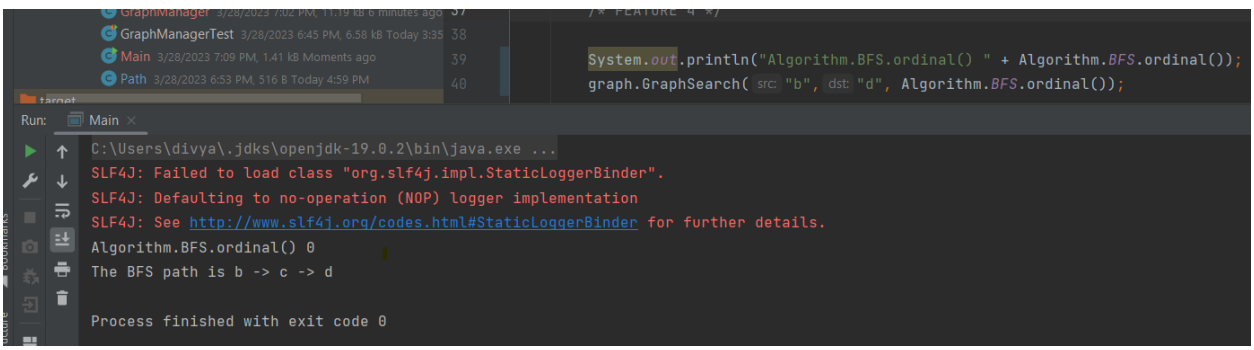
**ASSIGNMENT 2:**

**BFS:**

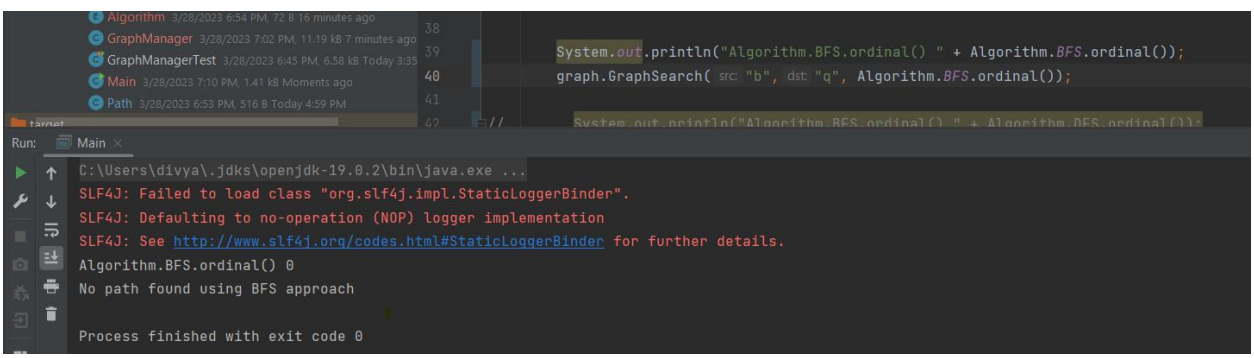
To run the BFS part of the assignment, run the Main.java file, which contains the below code.

```
graph.GraphSearch("b","d", Algorithm.BFS.ordinal());
```

The above code uses the enum values defined in **Algortithm.java** file to select the BFS approach to finding the path between source node (“b”) and destination node (“d”). We can modify the source node and destination node to test various combinations. The output will be as follows for a valid path:




The output will be as follows for an invalid path:

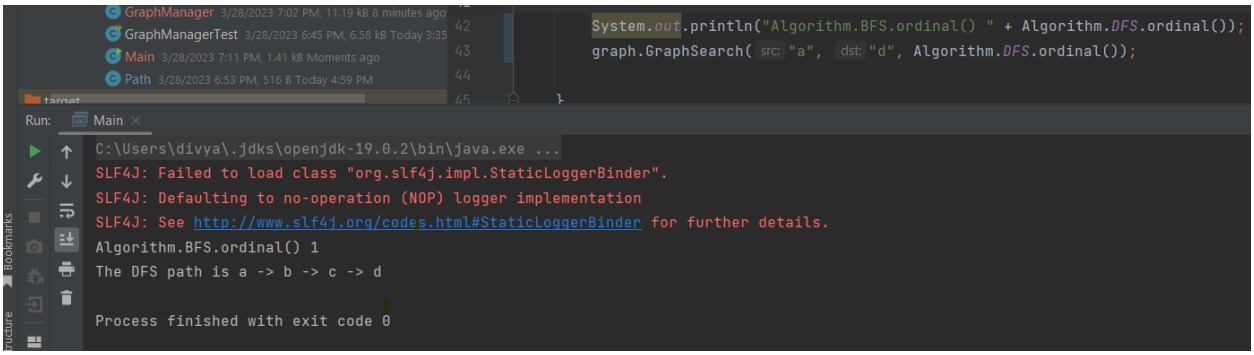


**DFS:**

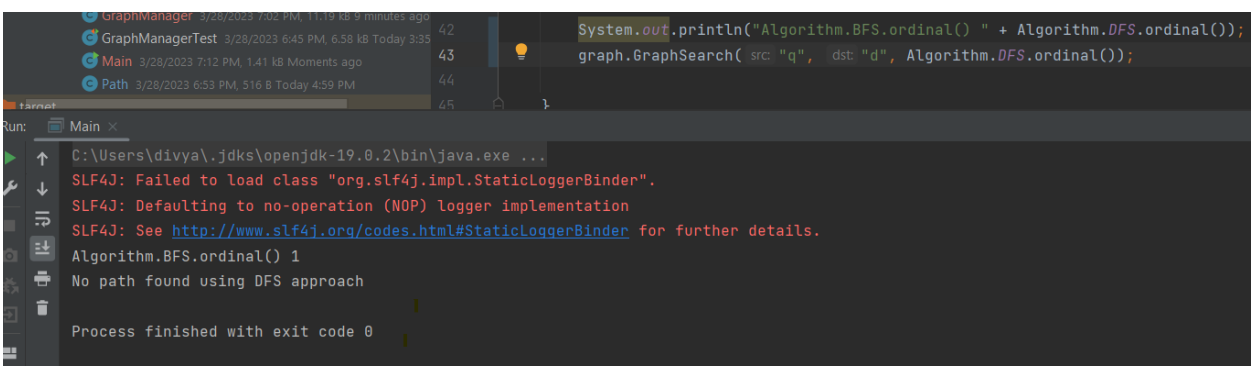
To run the DFS part of the assignment, run the Main.java file, which contains the below code.

**graph.GraphSearch("b","d", Algorithm.DFS.ordinal());**

The above code uses the enum values defined in **Algortithm.java** file to select the DFS approach to finding the path between source node ("a") and destination node ("d"). We can modify the source node and destination node to test various combinations. The output will be as follows for a valid path:




The output will be as follows for an invalid path:




The numerical equivalent for 0 is BFS, and that for 1 is DFS.

When we run both BFS and DFS together, as expected in the assignment, we get the below result for valid path.

```
System.out.println("Algorithm.BFS.ordinal() " + Algorithm.BFS.ordinal());
graph.GraphSearch( src: "b", dst: "d", Algorithm.BFS.ordinal());

System.out.println("Algorithm.BFS.ordinal() " + Algorithm.DFS.ordinal());
graph.GraphSearch( src: "a", dst: "d", Algorithm.DFS.ordinal());
```

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Algorithm.BFS.ordinal() 0
The BFS path is b -> c -> d
Algorithm.BFS.ordinal() 1
The DFS path is a -> b -> c -> d

Process finished with exit code 0
```

We get the below result when we give invalid paths:

```
System.out.println("Algorithm.BFS.ordinal() " + Algorithm.BFS.ordinal());
graph.GraphSearch( src: "q", dst: "d", Algorithm.BFS.ordinal());

System.out.println("Algorithm.BFS.ordinal() " + Algorithm.DFS.ordinal());
graph.GraphSearch( src: "y", dst: "d", Algorithm.DFS.ordinal());
```

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Algorithm.BFS.ordinal() 0
The BFS path is b -> c -> d
Algorithm.BFS.ordinal() 1
The DFS path is a -> b -> c -> d

Process finished with exit code 0
```

**CONCLUSION:**

When BFS was merged with the **main** branch, no conflicts were seen and I merged the two branches easily. When DFS was merged with the **main** branch, conflict was seen. So I updated DFS to include enum class **Algorithm** and then committed to DFS branch. Finally I merged DFS with main branch.

**LINKS:**

- Successful BFS Commit:
  - https://github.com/divyavijayakumar72/CSE_464_2023/actions/runs/4539727597
  - https://github.com/divyavijayakumar72/CSE_464_2023/commit/955ee8ee47b833835499c579adbfdfc0793d7568
- Successful DFS Commit:
  - https://github.com/divyavijayakumar72/CSE_464_2023/actions/runs/4539848320

- https://github.com/divyavijayakumar72/CSE_464_2023/commit/4d307c7e91c544685d2ae10134bbc2a270982918

- Successful BFS merge with main branch:
  - https://github.com/divyavijayakumar72/CSE_464_2023/actions/runs/4549037906
  - https://github.com/divyavijayakumar72/CSE_464_2023/commit/26d04865498da26bdb271fc874a6e63b9dfcf5e2

- Successful DFS merge with main branch:
  - https://github.com/divyavijayakumar72/CSE_464_2023/actions/runs/4550263242
  - https://github.com/divyavijayakumar72/CSE_464_2023/commit/2eae9e6dfdc02b198a7ce8b8537ef0f9103c675e