https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1

Course Project Part 2 For CSE 464

Features:

~NEW~

- removeNode – removes a node from the graph
  https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/4418c4f3846d64bde25bd4988cdf3aece77f3df5

- removeNodes – removes multiple nodes from the graph
  https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/4418c4f3846d64bde25bd4988cdf3aece77f3df5

- removeEdge – removes an edge between two nodes
  https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/844ccb1de8b97a2581674955bc59b1221942e60d

- graphSearch – can find a path between two nodes with your choice of breadth-first-search or depth-first-search
  BFS commit:
  https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/f80a58ff1d05d5258a7f5c27dc5d5f8c2ab593a2

  DFS commit:
  https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/bab54d776762ca919b28ca890d384bc873c3fc6b

  Merge of BFS and DFS branches:
  https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/ef27277814647c31ddd6f7d3bc575656a449c3ce

~OLD~

- parseGraph – parses DOT file
  https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/c59eeeacf8e0b01d00113c568aba2629dd9907a6

- addNode and addNodes – allows you to add a node to the graph
  https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/1f63bc43a9078c84aff55d94481f126420c750b7

- addEdge – allows you to add an edge between nodes
  https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/97e3ea4e13eebf2b652f8008196e3d73bfb7825a

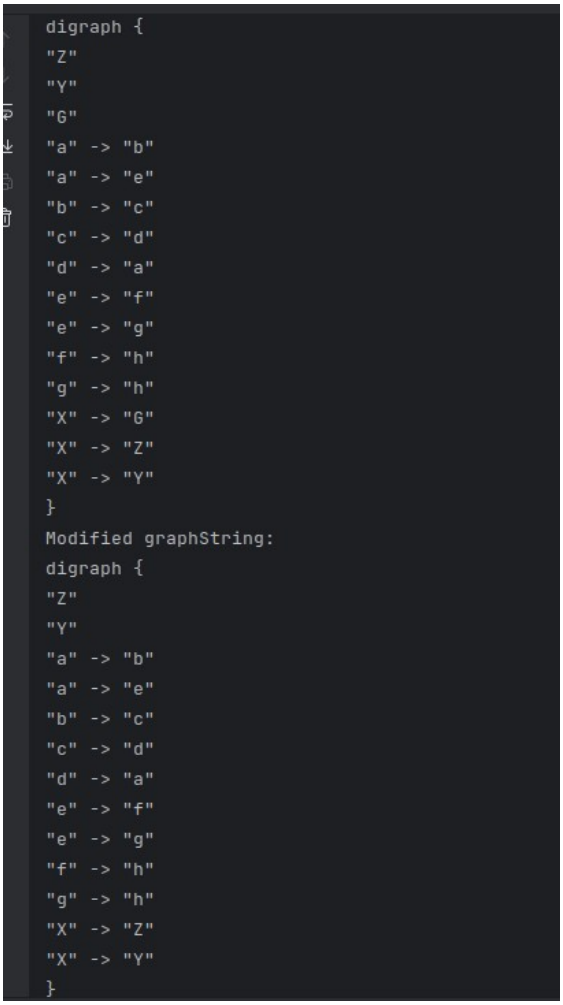
- outputDOTGraph – allows you to output a file of the graph in DOT format

- outputGraphics – allows you to output a png file of the graph https://github.com/Fonz-Hamilton/CSE-464-2024-Cahamil1/commit/844ccb1de8b97a2581674955bc59b1221942e60d

Examples

~NEW~
removeNode -

```
dotGraph.removeNode("G");
System.out.println(dotGraph.toString());
```



```
digraph {
"Z"
"Y"
"G"
"a" -> "b"
"a" -> "e"
"b" -> "c"
"c" -> "d"
"d" -> "a"
"e" -> "f"
"e" -> "g"
"f" -> "h"
"g" -> "h"
"X" -> "G"
"X" -> "Z"
"X" -> "Y"
}
Modified graphString:
digraph {
"Z"
"Y"
"a" -> "b"
"a" -> "e"
"b" -> "c"
"c" -> "d"
"d" -> "a"
"e" -> "f"
"e" -> "g"
"f" -> "h"
"g" -> "h"
"X" -> "Z"
"X" -> "Y"
}
```

removeNodes -

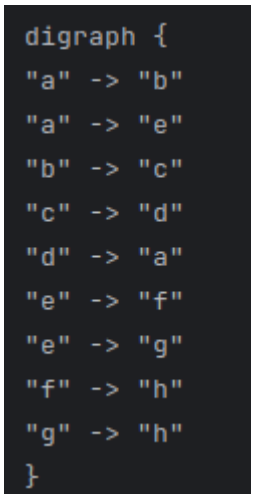
```
String[] nodesToAdd = {"Z", "X", "Y"};

dotGraph.removeNodes(nodesToAdd);
System.out.println(dotGraph.toString());
```

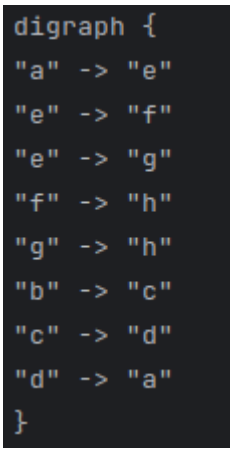
```
digraph {
"Z"
"Y"
"a" -> "b"
"a" -> "e"
"b" -> "c"
"c" -> "d"
"d" -> "a"
"e" -> "f"
"e" -> "g"
"f" -> "h"
"g" -> "h"
"X" -> "Z"
"X" -> "Y"
}
Graph rebuilt successfully.
digraph {
"a" -> "b"
"a" -> "e"
"b" -> "c"
"c" -> "d"
"d" -> "a"
"e" -> "f"
"e" -> "g"
"f" -> "h"
"g" -> "h"
"X" -> "Z"
"X" -> "Y"
}
```

```
Modified graphString:
digraph {
"a" -> "b"
"a" -> "e"
"b" -> "c"
"c" -> "d"
"d" -> "a"
"e" -> "f"
"e" -> "g"
"f" -> "h"
"g" -> "h"
"X" -> "Y"
}
Modified graphString:
digraph {
"a" -> "b"
"a" -> "e"
"b" -> "c"
"c" -> "d"
"d" -> "a"
"e" -> "f"
"e" -> "g"
"f" -> "h"
"g" -> "h"
}
```

removeEdge -

```
dotGraph.removeEdge("a","b");
System.out.println(dotGraph.toString());
```





graphSearch -

```
path = dotGraph.graphSearch(dotGraph.getNode("a"), dotGraph.getNode("h"),
DOTGraph.Algorithm.BFS);
System.out.println("BFS: \n" + path.printPath());
```

```
BFS:
a -> e -> f -> h
```

```
Path path = dotGraph.graphSearch(dotGraph.getNode("c"), dotGraph.getNode("h"),
DOTGraph.Algorithm.DFS);
System.out.println("DFS: \n" + path.printPath());
```

```
DFS:
c -> d -> a -> e -> f -> h
```

~OLD~

parseGraph -

```
DOTGraph dotGraph = new DOTGraph();
dotGraph.parseGraph("input.dot");
System.out.println(dotGraph.toString());addNodes -
String[] nodesToAdd = {"Z", "X", "Y"};
dotGraph.addNodes(nodesToAdd);
System.out.println(dotGraph.toString());
```

addNode -

```
String addedNode = "G";
dotGraph.addNode(addedNode);
System.out.println(dotGraph.toString());addEdge -
dotGraph.addEdge("X","Y");
System.out.println(dotGraph.toString());
dotGraph.addEdge("X","Z");
System.out.println(dotGraph.toString());
dotGraph.addEdge("X","G");
System.out.println(dotGraph.toString());
```

outputDOTGraph -

```
dotGraph.outputDOTGraph("output.dot");
```

outputGraphic -

```
dotGraph.outputGraphics("output","png");
```