GraphDot Project 3 - CSE 464

Bradyn Flahart

Repo: https://github.com/btflahar/CSE-464-btflahar-graph.git

mvn package command works and will properly build

There should be a total of 15 Tests that run when package is run

Java 17
Apache Maven
GraphViz

Is required

mvn clean package to start

Place DOT file in the root (input.dot)

The tests can be run on either
src/test/java/btflahar/asu/edu/graphDot/FinalDemoTest.java
Or src/main/java/btflahar/asu/edu/graphDot/GraphApp

They can be run by simply going to the respective code block (main for either file) and running the green triangle (run button) for either.

The example input.dot given is very simple

```
digraph G {
  a -> b;
  b -> c;
  c -> d;
  d -> a;
  a -> e;
  e -> f;
  e -> g;
  f -> h;
  g -> h;
}
```

The expected outputs after running the TestDemo would be a successful BFS, DFS, and Random Walk

These outputs would look like

BFS

BFS Execution
visiting a
visiting a -> b
visiting a -> e
visiting a -> b -> c
visiting a -> e -> f
visiting a -> e -> g
visiting a -> b -> c -> d
visiting a -> e -> f -> h
Final BFS path: a -> e -> f -> h

DFS

DFS Execution
visiting a
visiting a -> e
visiting a -> e -> g
visiting a -> e -> g -> h
Final DFS path: a -> e -> g -> h

RandomWalk

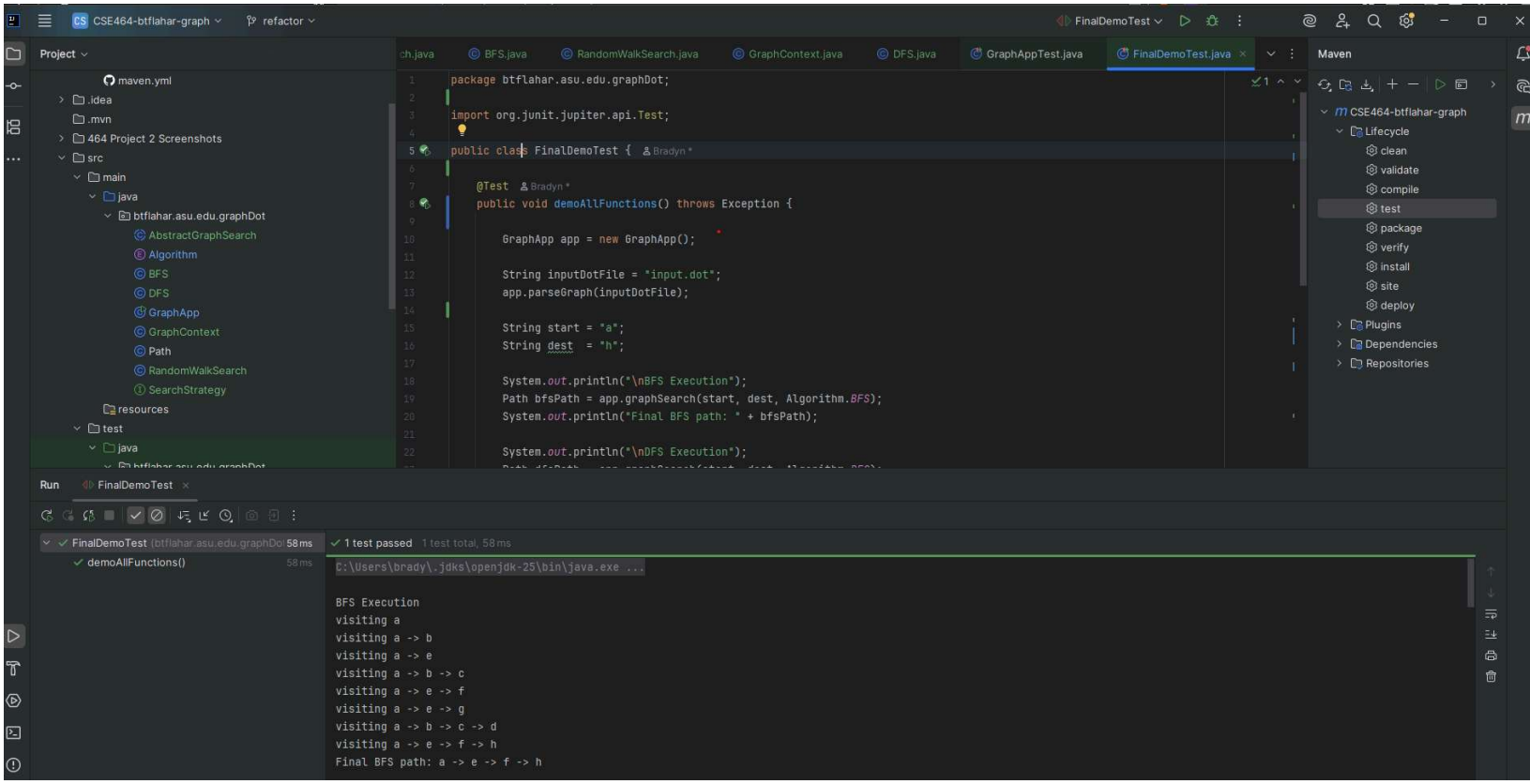
Random Walk output will vary but examples of it can be shown below
Random Walk Execution
visiting a
visiting a -> e
visiting a -> e -> f
visiting a -> e -> f -> h
Random Walk #1: a -> e -> f -> h
visiting a
visiting a -> e
visiting a -> e -> f
visiting a -> e -> f -> h
Random Walk #2: a -> e -> f -> h

visiting a
visiting a -> e
visiting a -> e -> f
visiting a -> e -> f -> h
Random Walk #3: a -> e -> f -> h
visiting a
visiting a -> b
visiting a -> b -> c
visiting a -> b -> c -> d
Random Walk #4: (dead end)
visiting a
visiting a -> b
visiting a -> b -> c
visiting a -> b -> c -> d
Random Walk #5: (dead end)
visiting a
visiting a -> b
visiting a -> b -> c
visiting a -> b -> c -> d
Random Walk #6: (dead end)
visiting a
visiting a -> e
visiting a -> e -> f
visiting a -> e -> f -> h
Random Walk #7: a -> e -> f -> h
visiting a
visiting a -> b
visiting a -> b -> c
visiting a -> b -> c -> d
Random Walk #8: (dead end)
visiting a
visiting a -> e
visiting a -> e -> f
visiting a -> e -> f -> h
Random Walk #9: a -> e -> f -> h
visiting a
visiting a -> e
visiting a -> e -> f
visiting a -> e -> f -> h
Random Walk #10: a -> e -> f -> h

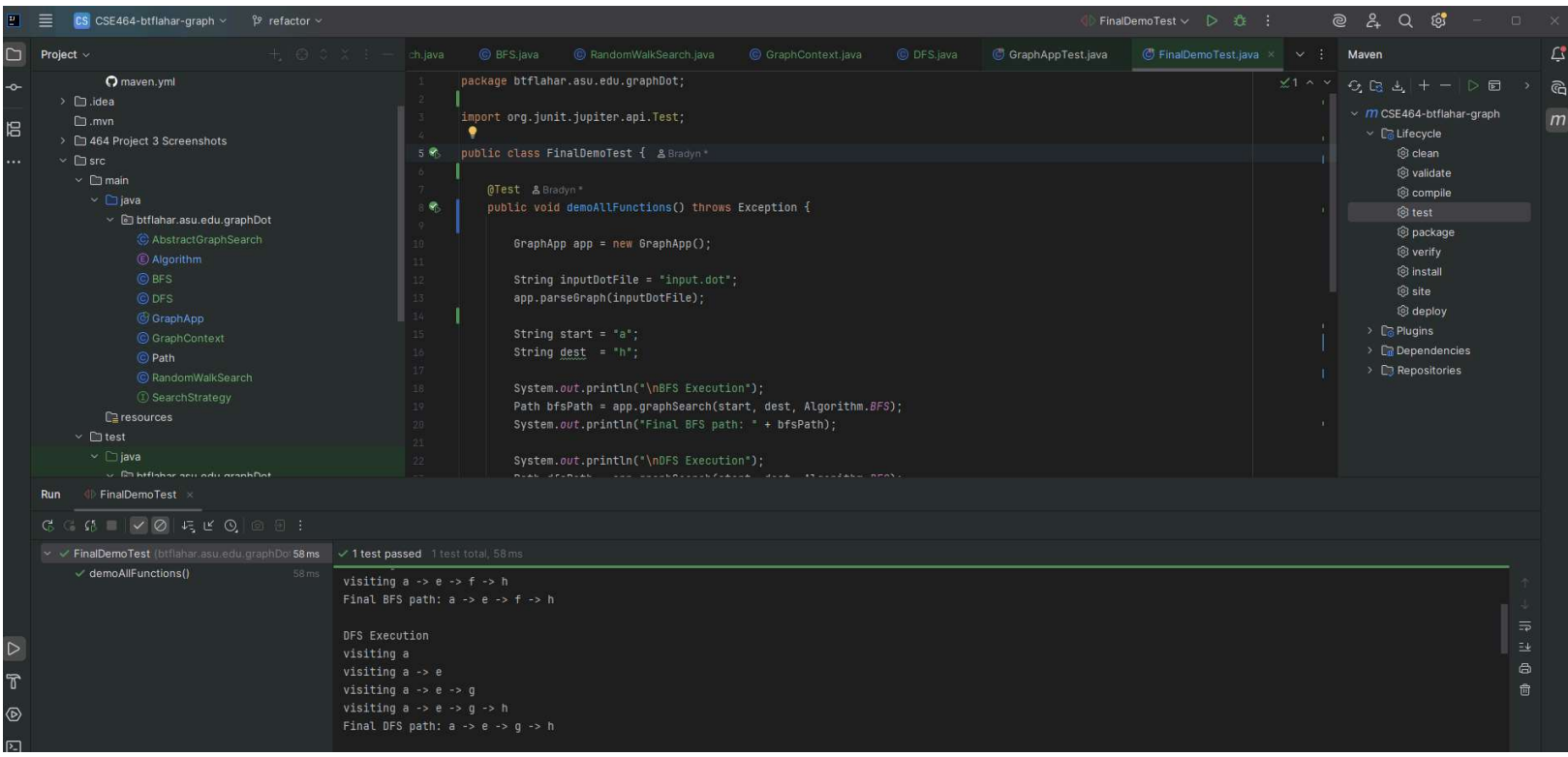
The desired start/end node are entered at
src/test/java/btflahar/asu/edu/graphDot/FinalDemoTest.java
At “Start node” and “End Node”

Enter the desired node for each.

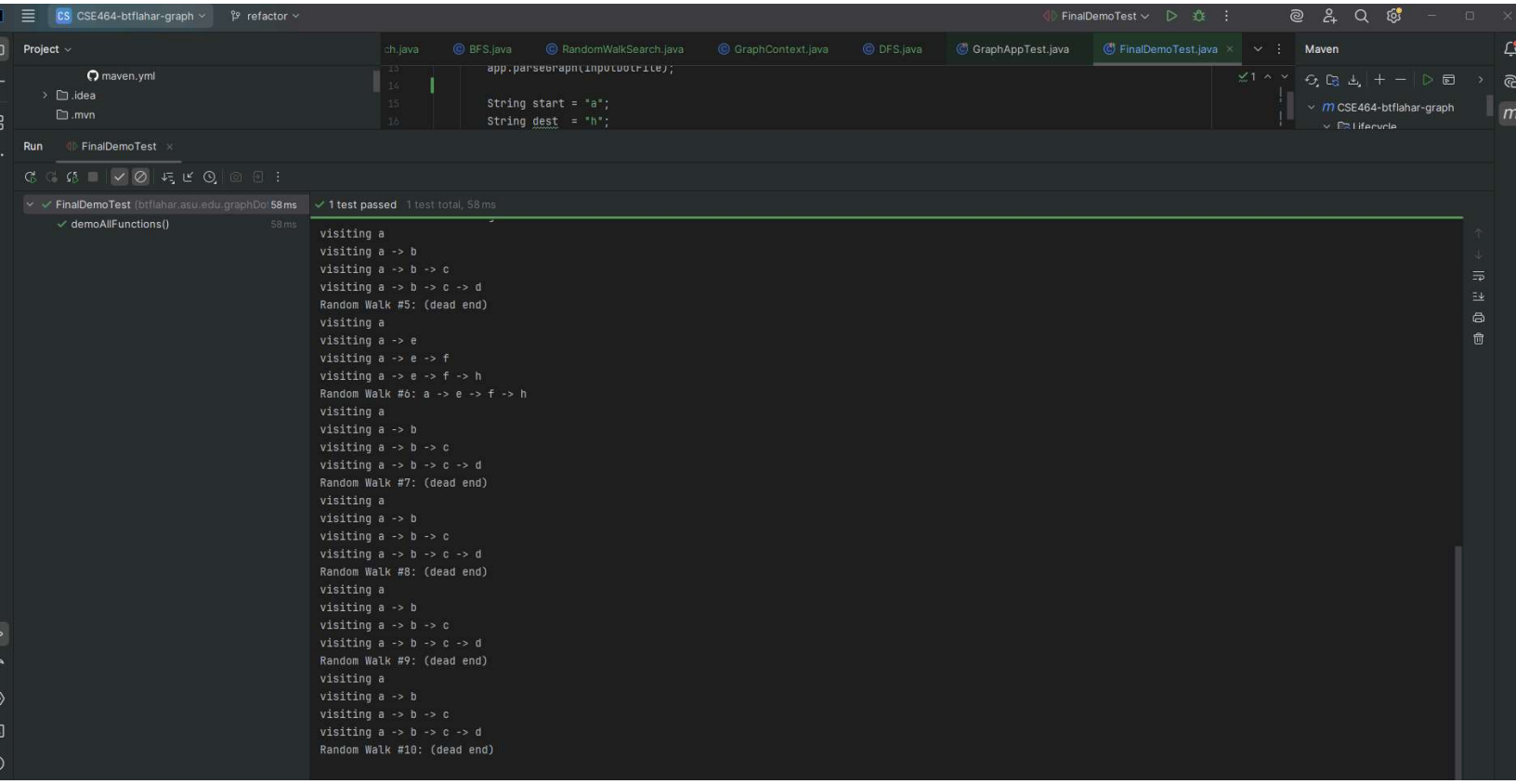
Here are images of each respective output
BFS

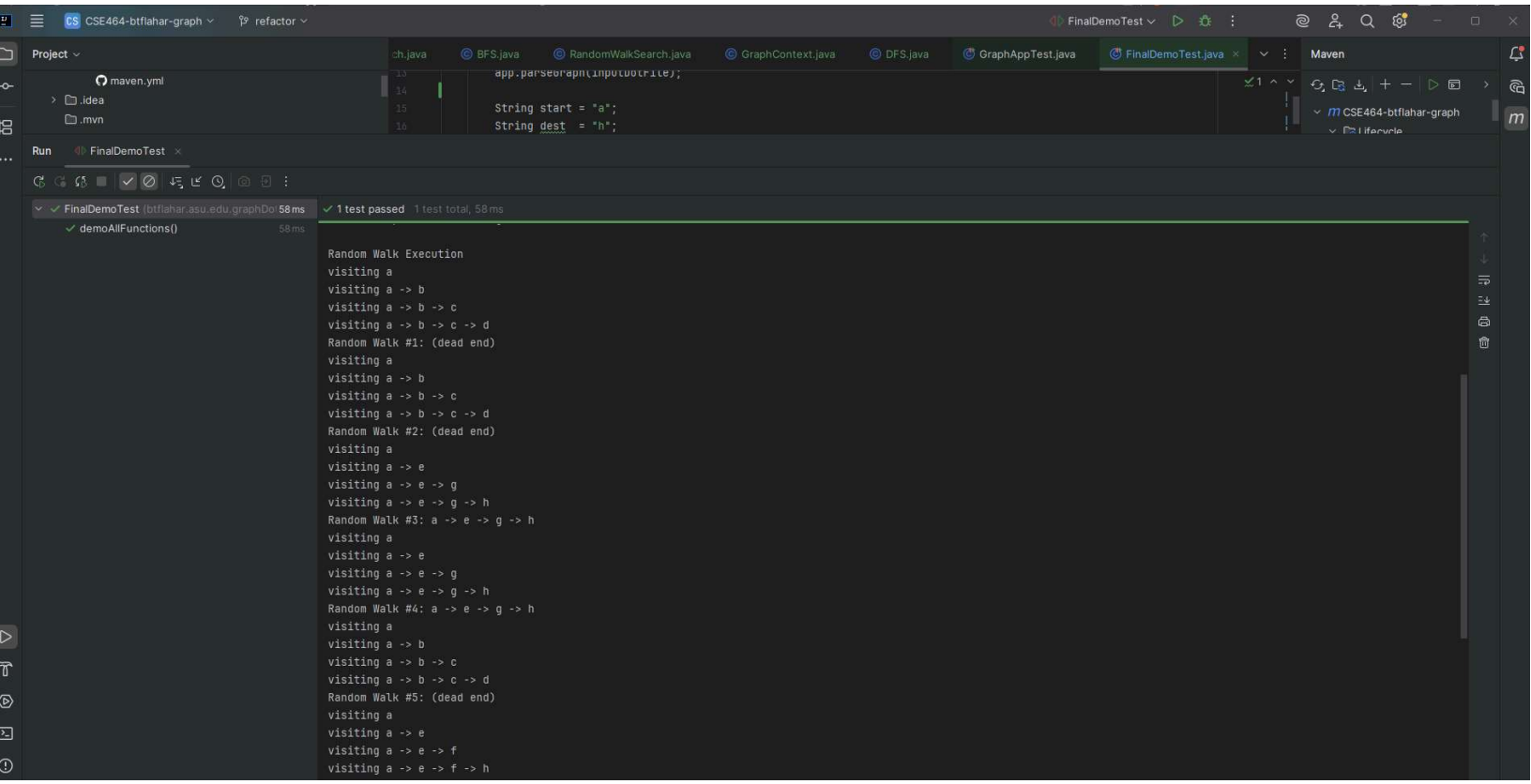


DFS



RandomWalk 1 and 2





Refactors

Refactor 1:
https://github.com/btflahar/CSE-464-btflahar-graph/commit/e8962f645620b6ae4eb5f32d2da2b372ad5d9924

Refactor 2:
https://github.com/btflahar/CSE-464-btflahar-graph/commit/beefdccc793258fc255ce5450c8a654ca54d0eec

Refactor 3:
https://github.com/btflahar/CSE-464-btflahar-graph/commit/233b72533b0bf5b03ffa80adefc6ecd81840fe01

Refactor 4:
https://github.com/btflahar/CSE-464-btflahar-graph/commit/ff2e1d0bc7ad6b810068208632a76644f2778b3f

Refactor 5:
https://github.com/btflahar/CSE-464-btflahar-graph/commit/2578141c98517c857962f4ff3d01077896e9f358

Last 2 Commits (one after pull request)

https://github.com/btflahar/CSE-464-btflahar-graph/commit/d76d07919a1cabe1e061409049e5f01124880205

https://github.com/btflahar/CSE-464-btflahar-graph/commit/58b773eb0feb12672a56e8ae41066897d1a9cac9 (after pull request.)