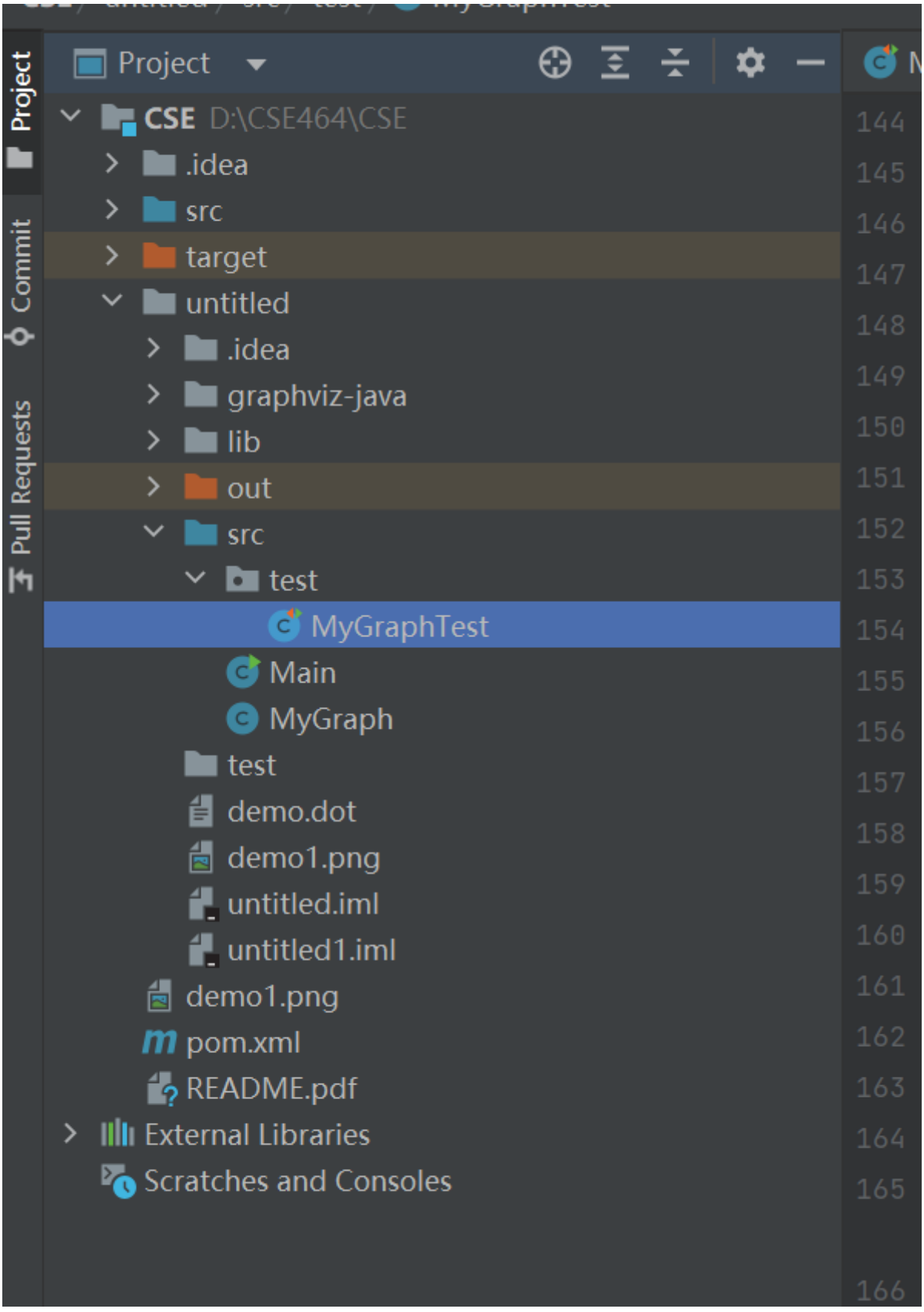
Structure of the project

https://github.com/TopKingHao/cse464p2



Step 1

After running

```
mvn package
```

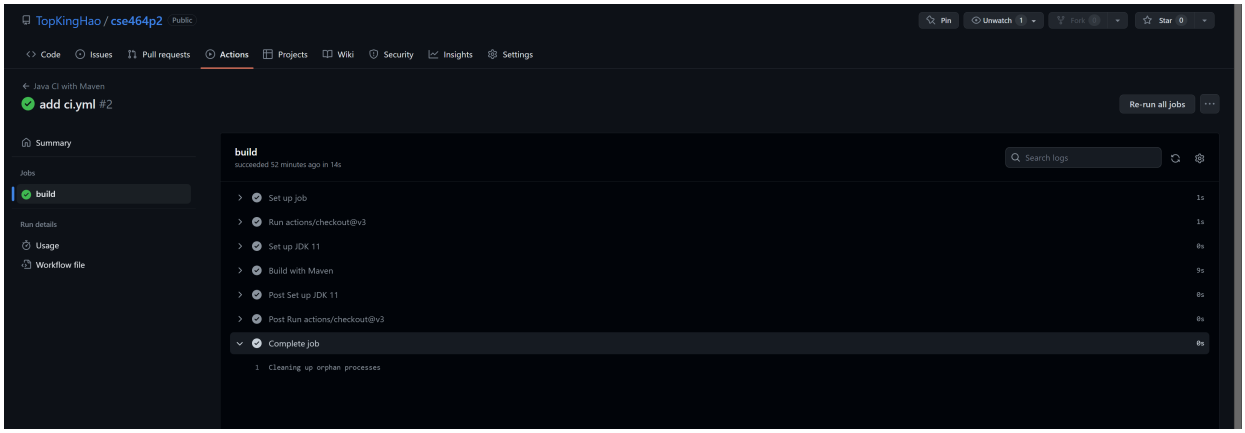
We have below results to show that building and testing could work.

```
xuanwu@Xuanwu-PC:~/test/cse464pp2$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------------< hzhen:cse464 >----------------------------
[INFO] Building cse464pp2 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/guru/nidi/graphviz-java/0.18.1/graphviz-java-
Downloaded from central: https://repo.maven.apache.org/maven2/guru/nidi/graphviz-java/0.18.1/graphviz-java-0
.7 kB at 46 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/guru/nidi/graphviz-java-parent/0.18.1/graphvi
t-0.18.1.pom
```

```
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running hzhen.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.014 s - in hzhen.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.0.2:jar (default-jar) @ cse464 ---
[INFO] Building jar: /home/xuanwu/test/cse464pp2/target/cse464-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.720 s
[INFO] Finished at: 2023-03-28T11:58:27-05:00
[INFO] ------------------------------------------------------------------------
```

Step 2

The work flow is exactly as expected



Step 3

For BFS, it is a good idea to use a priority queue to simulate the problem.

We try all possibilities for one node, stores them at the end of the queue and run into the next element.

```
no usages  TopKingHao
public Path GraphSearch(String src, String dst) {
    if (src == null || dst == null || !myGraph.containsVertex(src) || !myGraph.containsVertex(dst)) {
        return null;
    }

    Queue<String> queue = new LinkedList<>();
    Map<String, String> previousNodes = new HashMap<>();
    Set<String> visitedNodes = new HashSet<>();

    queue.add(src);

    while (!queue.isEmpty()) {
        String currentNode = queue.poll();

        if (currentNode.equals(dst)) {
            Path path = new Path();
            String pathNode = dst;
            while (pathNode != null) {
                path.addNode(pathNode);
                pathNode = previousNodes.get(pathNode);
            }
            Collections.reverse(path.getNodes());
            return path;
        }

        visitedNodes.add(currentNode);

        for (DefaultEdge edge : myGraph.edgesOf(currentNode)) {
            String neighbor = myGraph.getEdgeTarget(edge);
            if (!visitedNodes.contains(neighbor) && !queue.contains(neighbor)) {
                previousNodes.put(neighbor, currentNode);
                queue.add(neighbor);
            }
        }
    }

    return null;
}
```

Step 4

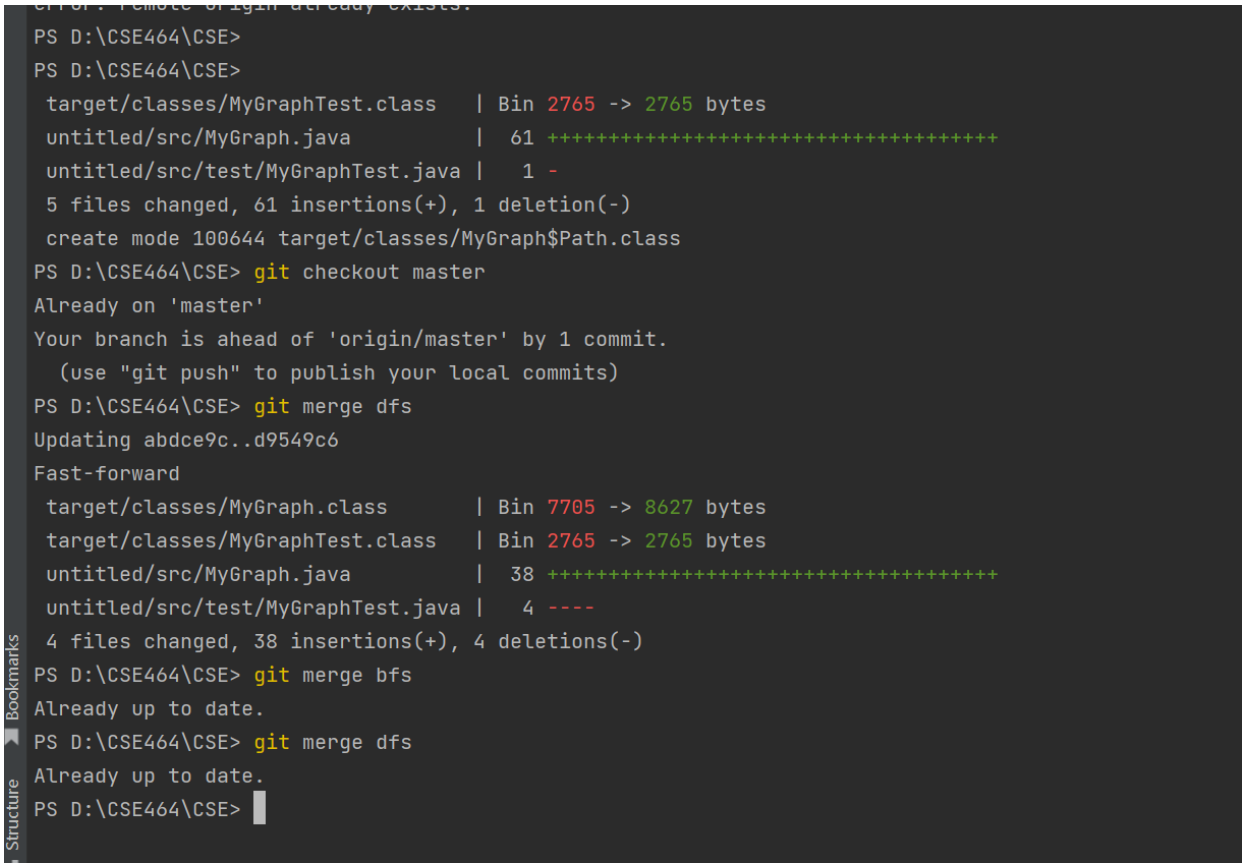
For DFS, we use a helper function to simulating recursive status.

The process wont stop until it reaches the end of one path, which means it will stop until all elements are out.

```java
    no usages  TopKingHao
    public Path GraphSearchDFS(String src, String dst) {
        if (src == null || dst == null || !myGraph.containsVertex(src) || !myGraph.containsVertex(dst)) {
            return null;
        }

        Set<String> visitedNodes = new HashSet<>();
        Map<String, String> previousNodes = new HashMap<>();

        if (dfsHelper(src, dst, visitedNodes, previousNodes)) {
            Path path = new Path();
            String pathNode = dst;
            while (pathNode != null) {
                path.addNode(pathNode);
                pathNode = previousNodes.get(pathNode);
            }
            Collections.reverse(path.getNodes());
            return path;
        }

        return null;
    }

    2 usages  TopKingHao
    private boolean dfsHelper(String currentNode, String dst, Set<String> visitedNodes, Map<String, String> previousNodes) {
        visitedNodes.add(currentNode);
        if (currentNode.equals(dst)) {
            return true;
        }
        for (DefaultEdge edge : myGraph.edgesOf(currentNode)) {
            String neighbor = myGraph.getEdgeTarget(edge);
            if (!visitedNodes.contains(neighbor)) {
                previousNodes.put(neighbor, currentNode);
                if (dfsHelper(neighbor, dst, visitedNodes, previousNodes)) {
                    return true;
                }
            }
        }
        return false;
    }

}
```

Step 5

Merged successfully


error: remote origin already exists.
PS D:\CSE464\CSE>
PS D:\CSE464\CSE>
target/classes/MyGraphTest.class | Bin 2765 -> 2765 bytes
untitled/src/MyGraph.java | 61 ++
untitled/src/test/MyGraphTest.java | 1 -
5 files changed, 61 insertions(+), 1 deletion(-)
create mode 100644 target/classes/MyGraph$Path.class
PS D:\CSE464\CSE> git checkout master
Already on 'master'
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
PS D:\CSE464\CSE> git merge dfs
Updating abdce9c..d9549c6
Fast-forward
target/classes/MyGraph.class | Bin 7705 -> 8627 bytes
target/classes/MyGraphTest.class | Bin 2765 -> 2765 bytes
untitled/src/MyGraph.java | 38 ++++++++++++++++++++++++++++++++++++++
untitled/src/test/MyGraphTest.java | 4 ----
4 files changed, 38 insertions(+), 4 deletions(-)
PS D:\CSE464\CSE> git merge bfs
Already up to date.
PS D:\CSE464\CSE> git merge dfs
Already up to date.
PS D:\CSE464\CSE>
Bookmarks
Structure