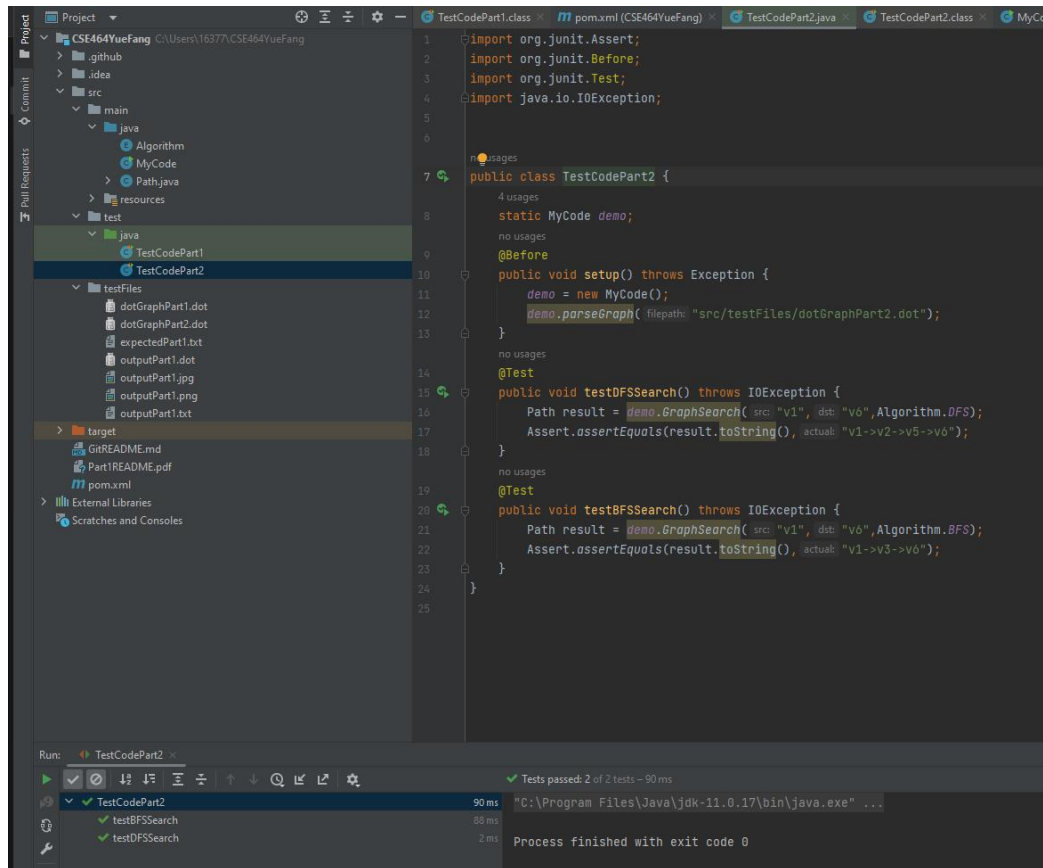


The screenshot displays an IDE environment with the following components:

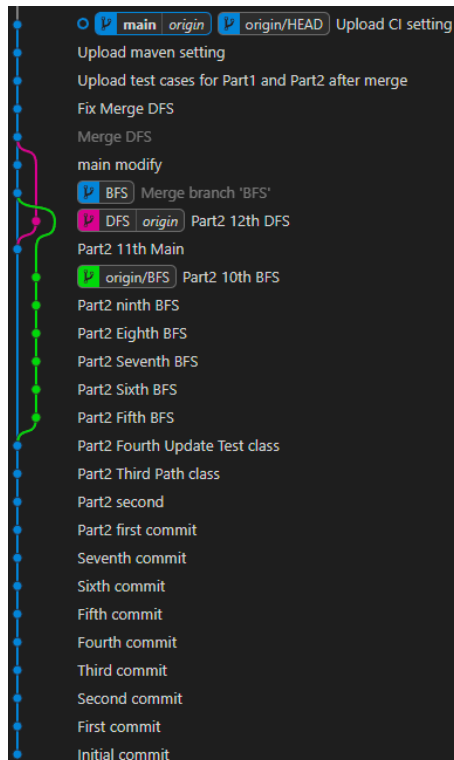
- Top Pane (Project Explorer):** Shows the project structure. The 'test' directory is expanded, revealing 'TestCodePart1' and 'TestCodePart2' under the 'java' subdirectory. The 'TestCodePart1' file is selected.
- Middle Pane (Source Editor):** Displays the source code of 'TestCodePart1.class'. The code includes imports for 'org.junit.jupiter.api.Assertions', 'org.junit.jupiter.api.BeforeEach', 'org.junit.jupiter.api.Test', and 'org.junit.jupiter.api.io.TempDir'. The class 'TestCodePart1' contains a 'MyCode' instance and several test methods: 'setUp()', 'testFeature1()', and 'testFeature2()'. The 'testFeature1()' method performs assertions on the size of the vertex set and the presence of an edge. The 'testFeature2()' method performs assertions on the size of the vertex set and the removal of nodes.
- Bottom Pane (Test Runner):** Shows the results of the test run. The test 'TestCodePart1' is listed with a duration of 417ms. Below it, four individual test methods are listed, all of which passed: 'testFeature1' (94ms), 'testFeature2' (4ms), 'testFeature3' (2ms), and 'testFeature4' (317ms). The total duration for all tests is 417ms. The status bar indicates 'Tests passed: 4 of 4 tests - 417ms'.



I got some problems when the first time I try to merge BFS and DFS, and I committed some useless code because of this, so I remove or modify these commits and the Workflows for them because these will influence my future work. If you find Workflow lost some runs, please do not feel strange. If this makes some problems for your view, sorry for that. Also, there are some problems with maven (forgot add path for test) and ci.yml (use wrong code, I should use "mvn test", but I use "mvn package"), so I upload a new version of them after I finish all the merges.

Following are the screen shot of my process:

Screen shot of my path graph:



Screen shot for set maven support and CI: (final commit name: **Part2 second**)

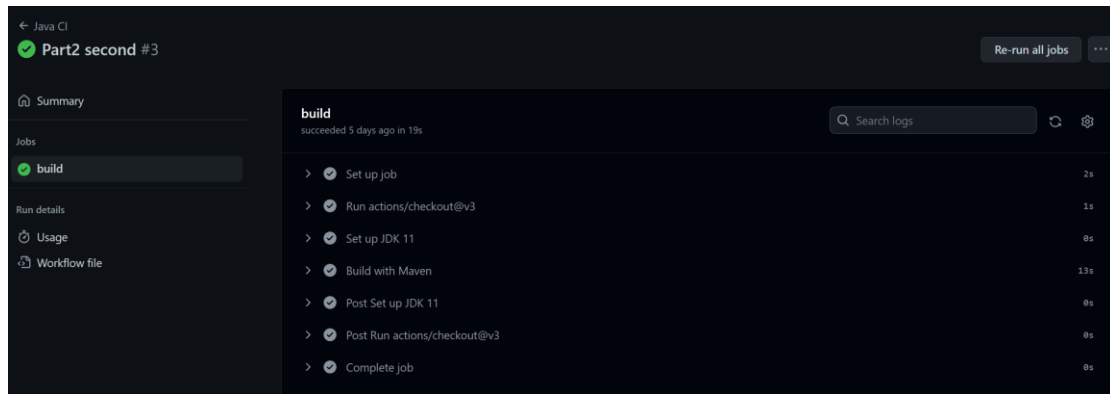


The first commit did not pass CI because of the path problem so I make some changes and do the second commit based on the first one. Second is the final result.

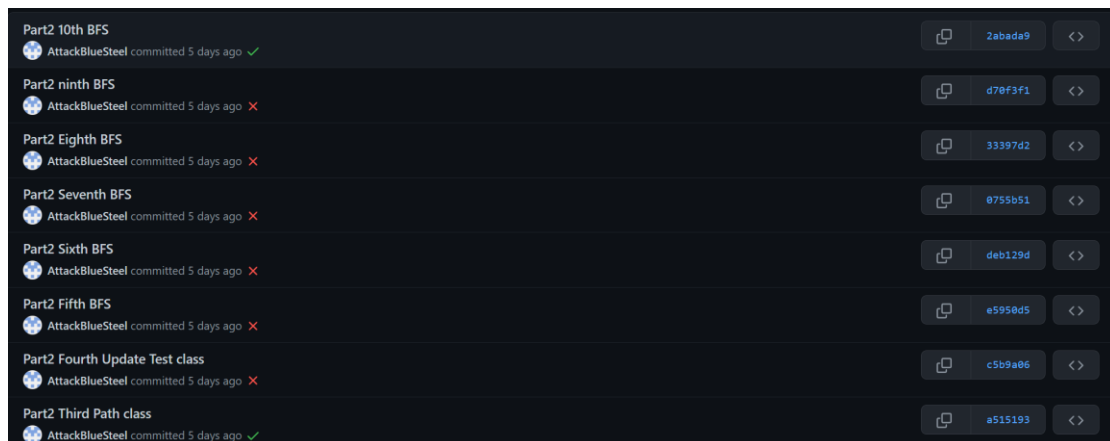
<https://github.com/AttackBlueSteel/CSE464YueFang/tree/a54d3f24786b2142a7722c28eaa5df65b16072e6>

| AttackBlueSteel Part2 second |                    |            | ✓ a54d3f2 5 days ago | 🕒 10 commits |
|------------------------------|--------------------|------------|----------------------|--------------|
| 📁 .github/workflows          | Part2 first commit | 5 days ago |                      |              |
| 📁 .idea                      | Part2 second       | 5 days ago |                      |              |
| 📁 src                        | Part2 second       | 5 days ago |                      |              |
| 📁 target/classes             | Part2 second       | 5 days ago |                      |              |
| 📄 README.md                  | Initial commit     | last month |                      |              |
| 📄 README.pdf                 | Part2 second       | 5 days ago |                      |              |
| 📄 pom.xml                    | Part2 second       | 5 days ago |                      |              |

No change for files, only add maven and CI

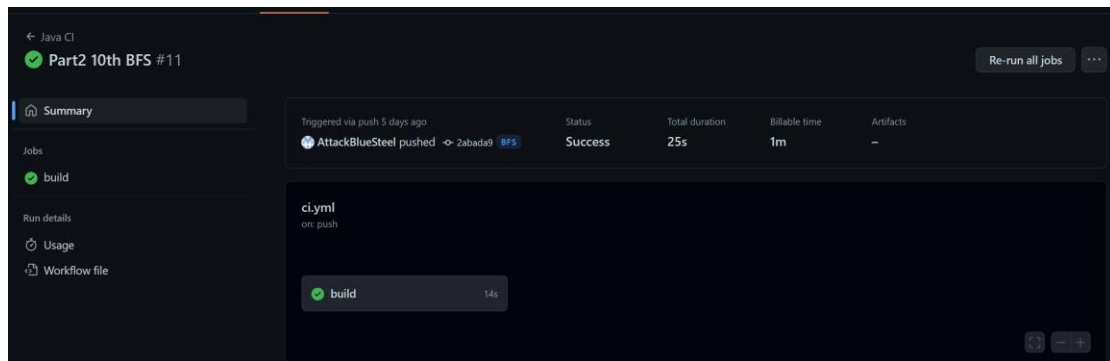


Screenshot for create BFS and make sure CI pass (Final commit name: Part2 10<sup>th</sup> BFS)



The change for this part is add the Path class, new test case, add some code, then create BFS branch and make sure the commit can pass CI. Because of some code problems, I changed some of my code and then make this part pass the CI. Part 2 10<sup>th</sup> BFS is the final result. The command I use in my git to create new branch is `git branch BFS`, then I changed my code in local repo and use `git checkout BFS` to switch to BFS branch and do `git commit` and `git push`. <https://github.com/AttackBlueSteel/CSE464YueFang/tree/BFS>

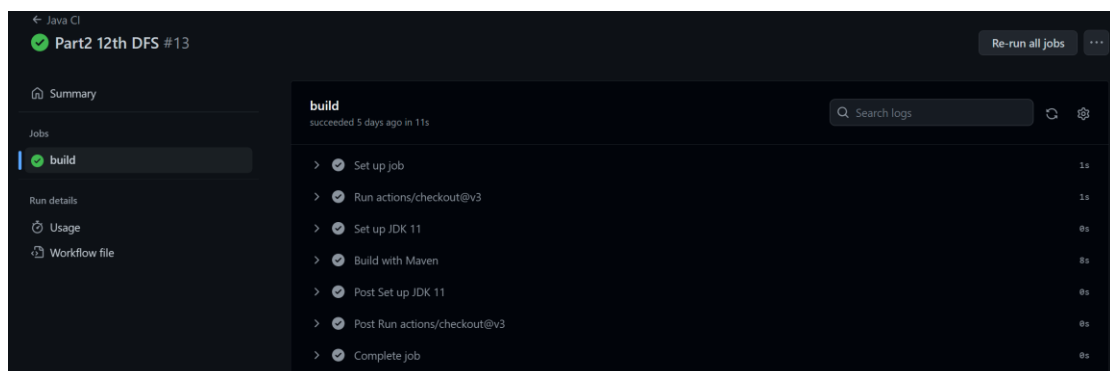
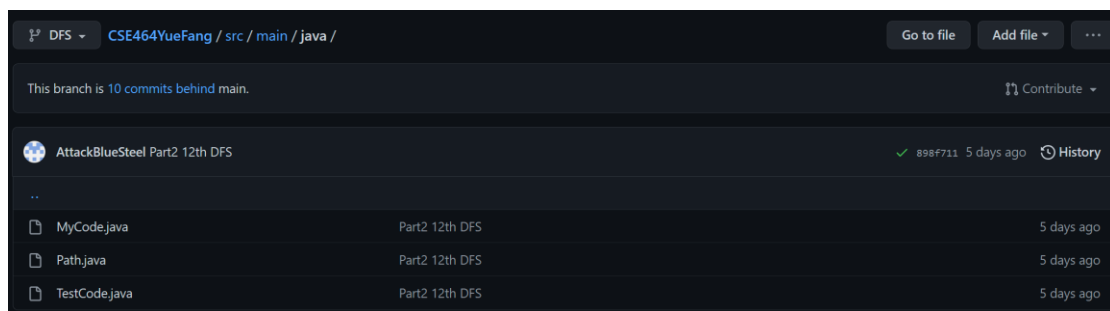




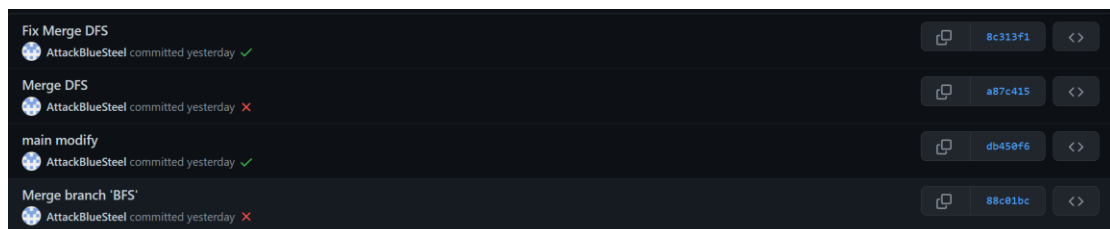
Screenshot for create DFS and make sure CI pass (Final commit name: Part2 12<sup>th</sup> BFS)



The change for this part is add the Path class, new test case, change some code, then create DFS branch and make sure the commit can pass CI, which is similar as the BFS part. Based on the experience on BFS, I only use one commit to finish this step. **The command I use in my git to create new branch is `git branch DFS`, then I changed my code in local repo and use `git checkout DFS` to switch to BFS branch and do git commit and git push.** <https://github.com/AttackBlueSteel/CSE464YueFang/tree/DFS>



Screenshot for merge BFS, DFS and make sure CI pass (Final commit name: Fix Merge DFS)



The change for this part is merge BFS to the main, and then merge DFS to the main. **The command I use is `git checkout main` to switch to main, and then `git merge BFS` to merge**

BFS to main and do similar merge code to DFS. I merge BFS first, and there is no conflict which is what I hope. But there is some problem with the Path class. Because of some actions I did in the past commit, GitHub think I may not want to keep the path class, so GitHub delete the path when I try to merge BFS. I do some changes and then make a new commit [main modify](#) and pass the CI after I merge BFS successfully. Then I merge DFS use similar way, but I got a conflict (see screen shot). So I open my code in VScode and accept both changes and write an enum Algorithm to make sure the program can switch the function. Then I do the commit, which is called Merge DFS. But I forgot to change some code in my test class, and this make this commit cannot pass CI. So I fix the problem and do another commit, which is called [Fix Merge DFS](#). Which should be the last commit for the coding part.

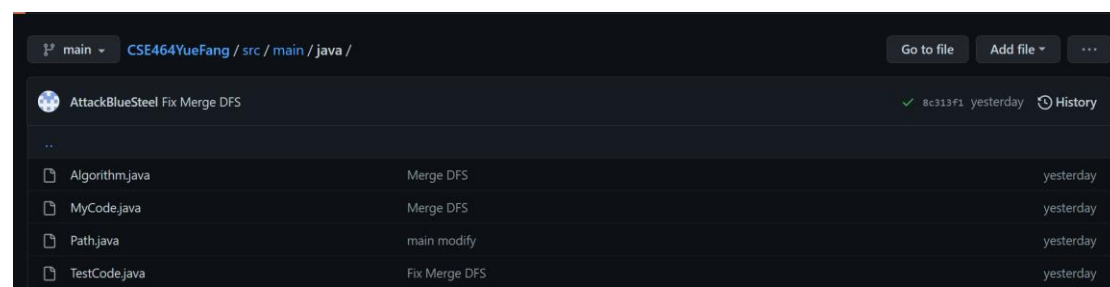
<https://github.com/AttackBlueSteel/CSE464YueFang/tree/8c313f16dadb72f67c57f4e0b7d5d50cb0001233>

```
16377@DESKTOP-7B49EA7 MINGW64 ~/CSE464YueFang (DFS)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

16377@DESKTOP-7B49EA7 MINGW64 ~/CSE464YueFang (main)
$ git merge BFS
Merge made by the 'ort' strategy.
 pom.xml | 5 +++++
src/main/java/MyCode.java | 30 ++++++++++++++++++++++-----
 2 files changed, 33 insertions(+), 2 deletions(-)

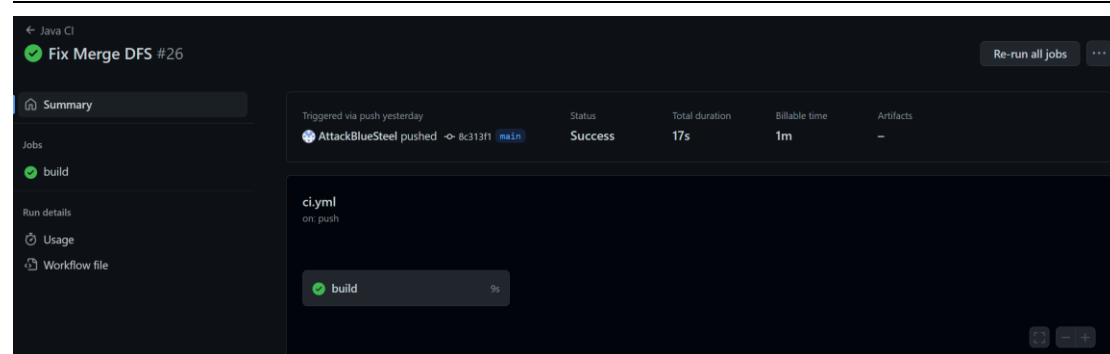
16377@DESKTOP-7B49EA7 MINGW64 ~/CSE464YueFang (main)
$ git merge DFS
Auto-merging src/main/java/MyCode.java
CONFLICT (content): Merge conflict in src/main/java/MyCode.java
Automatic merge failed; fix conflicts and then commit the result.

16377@DESKTOP-7B49EA7 MINGW64 ~/CSE464YueFang (main|MERGING)
$
```



The screenshot shows the GitHub repository interface for 'AttackBlueSteel Fix Merge DFS'. It lists several files: Algorithm.java, MyCode.java, Path.java, and TestCode.java, each with a commit message and a timestamp of 'yesterday'.

| File           | Commit Message | Timestamp |
|----------------|----------------|-----------|
| Algorithm.java | Merge DFS      | yesterday |
| MyCode.java    | Merge DFS      | yesterday |
| Path.java      | main modify    | yesterday |
| TestCode.java  | Fix Merge DFS  | yesterday |

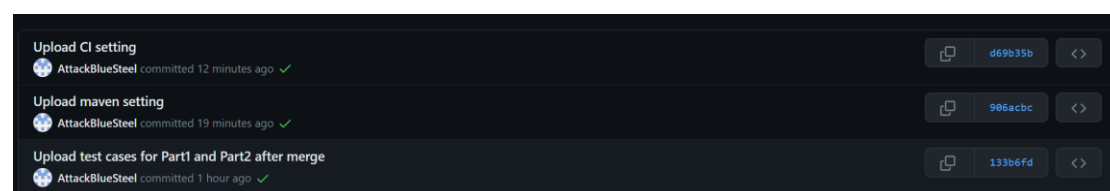


The screenshot shows the GitHub Actions workflow run for 'Fix Merge DFS #26'. It displays the summary, jobs, and run details. The workflow was triggered by a push to the main branch and completed successfully.

Summary: Triggered via push yesterday, Status: Success, Total duration: 17s, Billable time: 1m.

Jobs: build (Success, 9s).

This is the final version of my work:



The screenshot shows the GitHub repository commit history. It lists three commits: 'Upload CI setting', 'Upload maven setting', and 'Upload test cases for Part1 and Part2 after merge', all committed by 'AttackBlueSteel'.

| Commit Message                                    | Committed By    | Committed Time           |
|---|-----------------|--------------------------|
| Upload CI setting                                 | AttackBlueSteel | committed 12 minutes ago |
| Upload maven setting                              | AttackBlueSteel | committed 19 minutes ago |
| Upload test cases for Part1 and Part2 after merge | AttackBlueSteel | committed 1 hour ago     |

After I finish the merge, I found some error for my test cases ([I delet the test case for part1,](#)

and use a new test case for part2, but seems like we need both, so I add the test case for part1 back), maven (forgot to add test path) and CI (use wrong command, I should use mvn test, but I use mvn package). So I upload a new version of them to make sure the program is good to run and easier to be viewed.

<https://github.com/AttackBlueSteel/CSE464YueFang/tree/d69b35ba4649af1f0b36248d5767ea9f7419b698>

The screenshot shows a GitHub repository page for 'AttackBlueSteel Upload CI setting'. At the top, there's a notification that the main branch isn't protected. Below that, a commit history table lists several files: .github/workflows, .idea, src, target, GitREADME.md, Part1README.pdf, and pom.xml, each with a description of the upload and a timestamp. The bottom part of the image shows a CI/CD workflow run for 'Upload CI setting #29', which was triggered by a push to the main branch. The workflow status is 'Success' with a total duration of 34s. A sidebar on the left shows the workflow file 'ci.yml' and a 'build' job that took 23s.

```
3-3.0.0.jar (12 kB at 351 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/common-junit4/3.0.0/common-junit
4-3.0.0.jar (26 kB at 455 kB/s)
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running TestCodePart1
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.354 s - in TestCodePart1
[INFO] Running TestCodePart2
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in TestCodePart2
[INFO] Results:
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ CSE464YueFang ---
[INFO] Building jar: C:\Users\16377\CSE464YueFang\target\CSE464YueFang-1.0-SNAPSHOT.jar
[INFO] Copying org.example:CSE464YueFang:pom:1.0-SNAPSHOT to project local repository
[INFO] Copying org.example:CSE464YueFang:jar:1.0-SNAPSHOT to project local repository
[INFO] Copying org.example:CSE464YueFang:pom:consumer:1.0-SNAPSHOT to project local repository
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.577 s
[INFO] Finished at: 2023-03-28T20:56:15-07:00
[INFO] -----

C:\Users\16377\CSE464YueFang>
```

---

The code I submit on canvas will have three folders for three branches on GitHub, they are main, BFS, and DFS. Please choose the correct folder, thank you.