

Graph Traversal Framework

Overview

This project implements a graph traversal framework in Java, supporting the following traversal algorithms:

- **Breadth-First Search (BFS)**
- **Depth-First Search (DFS)**
- **Random Walk**

The framework includes:

1. Graph parsing from DOT files.
 2. Dynamic traversal strategy selection using the Strategy Pattern.
 3. Exporting graph data as DOT files or PNG visualizations.
 4. Continuous Integration (CI) setup via GitHub Actions for build and test automation.
-

Prerequisites

Required Tools

- **Java Development Kit (JDK) 11 or later**
- **Apache Maven** (for build and dependency management)
- **Git** (for version control)

External Libraries

- **JGraphT**: For graph representation and traversal.
- **Graphviz**: For exporting visualizations.

Installation

Run the following command to install dependencies:

```
mvn clean install
```

Setup Instructions

Clone the Repository:

```
git clone <your-repo-url>
cd <your-repo-folder>
```

Build the Project: Compile the codebase using Maven:

```
mvn package
```

1. Run the Code:

Breadth-First Search (BFS):

```
java -cp target/graph-traversal-framework.jar GraphParser
src/test/resources/sample.dot BFS
```

○

Depth-First Search (DFS):

```
java -cp target/graph-traversal-framework.jar GraphParser
src/test/resources/sample.dot DFS
```

○

Random Walk:

```
java -cp target/graph-traversal-framework.jar GraphParser
src/test/resources/sample_random_walk.dot RANDOM_WALK
```

Graph Traversals

Input Example

A DOT file for the graph:

```
digraph G {
```

```
A -> B;  
B -> C;  
C -> D;  
A -> D;  
}
```

Expected Outputs

BFS: Command:

```
java -cp target/graph-traversal-framework.jar GraphParser  
src/test/resources/sample.dot BFS
```

Output:

Path: A -> B -> C -> D

DFS: Command:

```
java -cp target/graph-traversal-framework.jar GraphParser  
src/test/resources/sample.dot DFS
```

Output:

Path: A -> D

Random Walk: Command:

```
java -cp target/graph-traversal-framework.jar GraphParser  
src/test/resources/sample_random_walk.dot RANDOM_WALK
```

Output:

Path: A -> E -> F -> H

Testing

Run the unit tests:

```
mvn test
```

Expected Output:

- All tests should pass successfully, verifying graph parsing, traversals, and graph export.

Continuous Integration (CI)

Setup

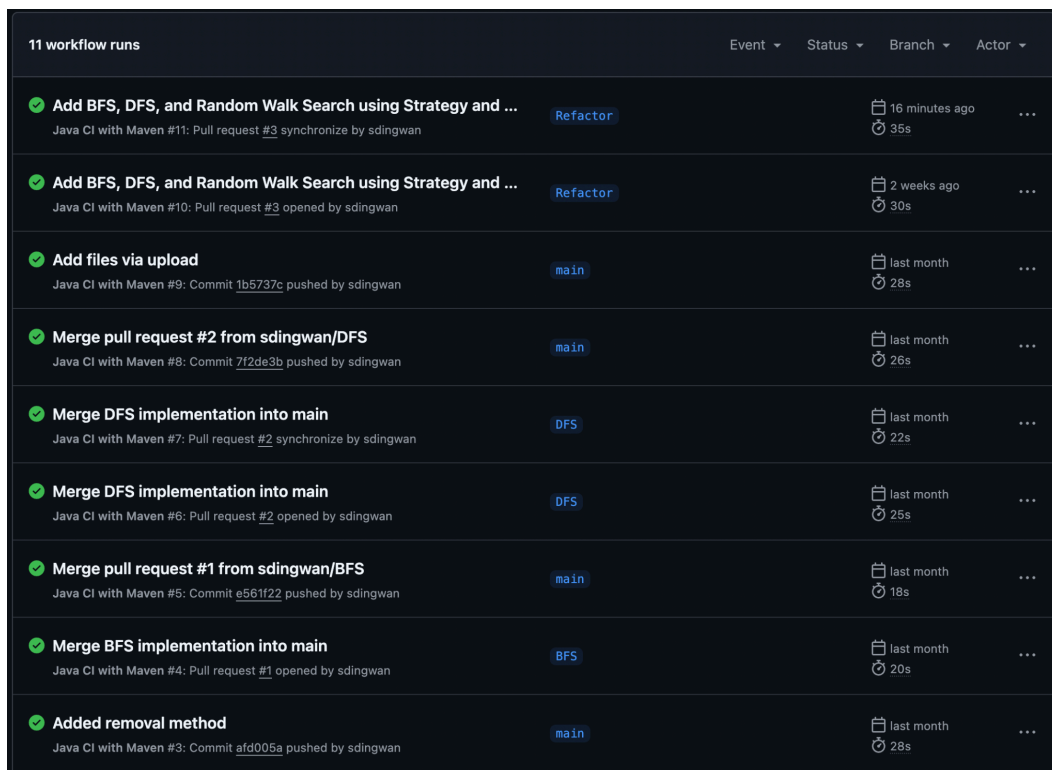
This project uses **GitHub Actions** for CI. The workflow file (`.github/workflows/maven.yml`) automates:

1. Code compilation using Maven.
2. Running unit tests for validation.

Results

Below are screenshots of:

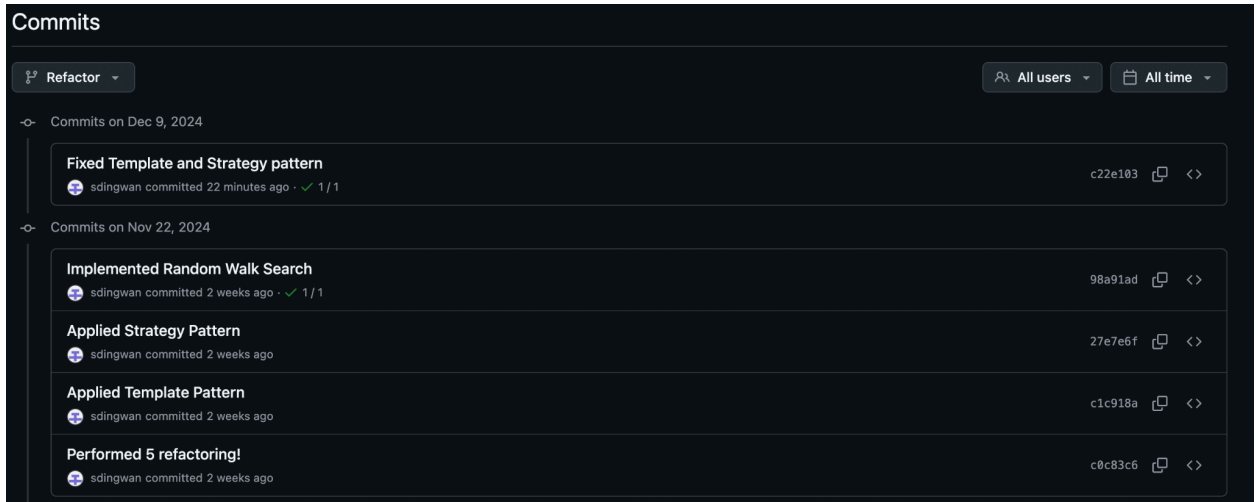
1. Successful CI Workflow results.



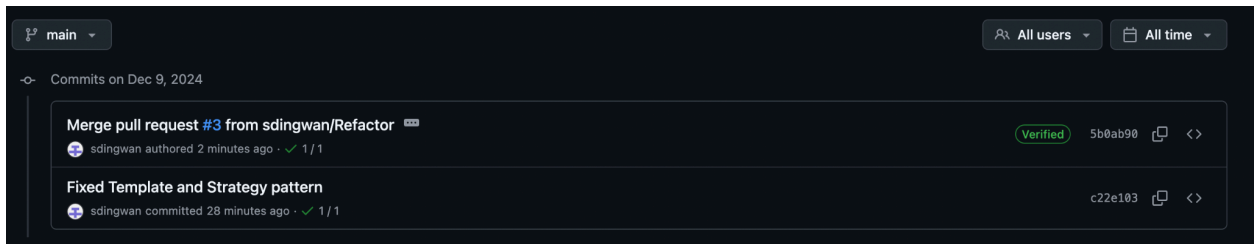
The screenshot displays a list of 11 workflow runs in a dark-themed interface. Each run is marked with a green checkmark, indicating success. The runs include details such as the workflow name, the event that triggered it (e.g., pull request, commit), the branch, and the duration. The workflow names are truncated, but they all relate to the project's CI process.

11 workflow runs	Event	Status	Branch	Actor
✓ Add BFS, DFS, and Random Walk Search using Strategy and ... Java CI with Maven #11: Pull request #3 synchronize by sdngwan	Refactor	16 minutes ago	35s	...
✓ Add BFS, DFS, and Random Walk Search using Strategy and ... Java CI with Maven #10: Pull request #3 opened by sdngwan	Refactor	2 weeks ago	30s	...
✓ Add files via upload Java CI with Maven #9: Commit 1b5737c pushed by sdngwan	main	last month	28s	...
✓ Merge pull request #2 from sdngwan/DFS Java CI with Maven #8: Commit 7f2de3b pushed by sdngwan	main	last month	26s	...
✓ Merge DFS implementation into main Java CI with Maven #7: Pull request #2 synchronize by sdngwan	DFS	last month	22s	...
✓ Merge DFS implementation into main Java CI with Maven #6: Pull request #2 opened by sdngwan	DFS	last month	25s	...
✓ Merge pull request #1 from sdngwan/BFS Java CI with Maven #5: Commit e561f22 pushed by sdngwan	main	last month	18s	...
✓ Merge BFS implementation into main Java CI with Maven #4: Pull request #1 opened by sdngwan	BFS	last month	20s	...
✓ Added removal method Java CI with Maven #3: Commit afd005a pushed by sdngwan	main	last month	28s	...

2. Commits showing feature implementation.



3. Merge confirmation.



Graph Export Examples

The framework allows exporting graphs to:

DOT files: Example command:

```
java -cp target/graph-traversal-framework.jar GraphParser  
src/test/resources/sample.dot OUTPUT_DOT src/test/resources/output.dot
```

•

PNG visualizations: Example command:

```
java -cp target/graph-traversal-framework.jar GraphParser  
src/test/resources/sample.dot OUTPUT_PNG src/test/resources/output.png
```

Example Output from Running Tests

✓ ✓ GraphParserTest	1 sec 572 ms
✓ testAddMultipleNodes()	19 ms
✓ testParseGraph()	5 ms
✓ testOutputDOTGraph()	2 ms
✓ testAddEdge()	
✓ testAddNode()	
✓ testGraphSearchStrategyPattern()	3 ms
✓ testGraphSearchBFS()	1 ms
✓ testGraphSearchDFS()	
✓ testRemoveEdge()	
✓ testRemoveNode()	
✓ testOutputGraphics()	1 sec 540 ms
✓ testGraphSearchRandomWalk()	2 ms
✓ testRemoveMultipleNodes()	

```

Random Walk Test Run 4:
Initializing traversal. Starting at node: A
visiting Path{nodes=[A, E]}
visiting Path{nodes=[A, E, F]}
visiting Path{nodes=[A, E, F, H]}
Random walk succeeded. Path: A -> E -> F -> H

Random Walk Test Run 5:
Initializing traversal. Starting at node: A
visiting Path{nodes=[A, E]}
visiting Path{nodes=[A, E, F]}
visiting Path{nodes=[A, E, F, H]}
Random walk succeeded. Path: A -> E -> F -> H

[INFO] Tests run: 13, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.932 s - in GraphParserTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 13, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ CSE-464-2024-sdingwan ---
[INFO] Building jar: /Users/saurabhdhingwani/IdeaProjects/CSE-464-2024-sdingwan/target/CSE-464-2024-sdingwan-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.443 s

```

GitHub Links

- **Repository:** <https://github.com/sdingwan/CSE-464-2024-sdingwan>
 - **Commit for BFS:** <https://github.com/sdingwan/CSE-464-2024-sdingwan/commits/BFS>
 - **Commit for DFS:** <https://github.com/sdingwan/CSE-464-2024-sdingwan/commits/DFS>
 - **Commit for Random Walk:**
<https://github.com/sdingwan/CSE-464-2024-sdingwan/commits/Refactor>
 - **CI Workflow:**
<https://github.com/sdingwan/CSE-464-2024-sdingwan/actions/workflows/maven.yml>
 - **Merge:**
<https://github.com/sdingwan/CSE-464-2024-sdingwan/actions/runs/12247833629>
-