A. Prerequisites

• Java Development Kit (JDK) 1.8 or later

• Apache Maven (Maven is used for building and managing dependencies)

B. Installing

1. Clone the repository to your local machine: git clone https://github.com/welldonepig/CSE-464-2023-tchen128.git

2. Navigate to the project directory: cd project-directory

3. Compile the project using Maven: mvn clean package

4. Run package command: mvn package

5. Running the Application: After you have successfully built the project, you can run the application using the following command:
mvn exec:java -Dexec.mainClass="org.parser.GraphParser"

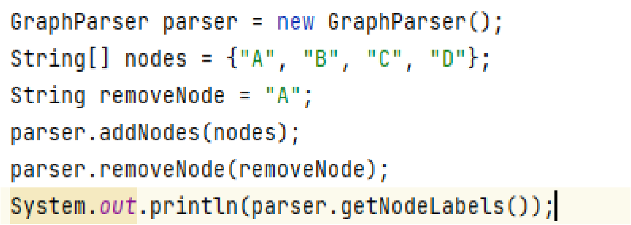
C. Instruction And Example

1. Run these test cases in **GraphParserTest.java** to test 3 new API

testRemoveNode
testRemoveNonExistentNode
testRemoveNodes
testRemoveNonExistentNodeInListNodes
removeEdge
testRemoveNonExistentEdge
testRemoveNonExistentEdgeBetweenExistingNodes

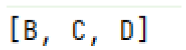
Expect all test case passed

- removeNode(String label)
Input:



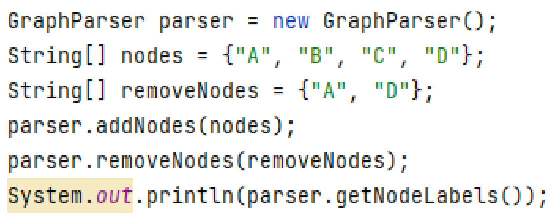
```
GraphParser parser = new GraphParser();
String[] nodes = {"A", "B", "C", "D"};
String removeNode = "A";
parser.addNodes(nodes);
parser.removeNode(removeNode);
System.out.println(parser.getNodeLabels());
```

Output:



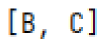
```
[B, C, D]
```

- removeNodes(String[] labels)
Input:



```
GraphParser parser = new GraphParser();
String[] nodes = {"A", "B", "C", "D"};
String[] removeNodes = {"A", "D"};
parser.addNodes(nodes);
parser.removeNodes(removeNodes);
System.out.println(parser.getNodeLabels());
```

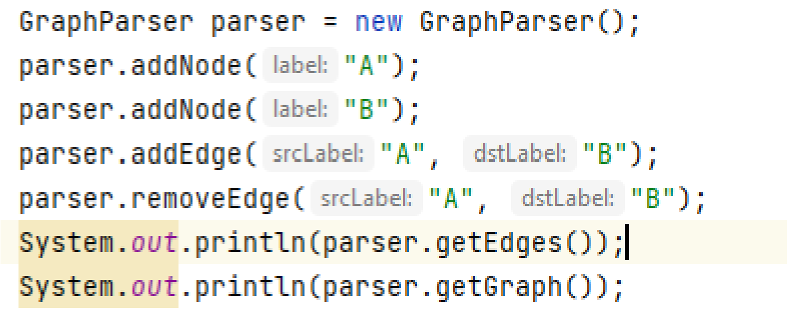
Output:



```
[B, C]
```

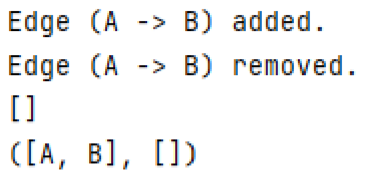
- removeEdge(String srcLabel, String dstLabel)

Input:



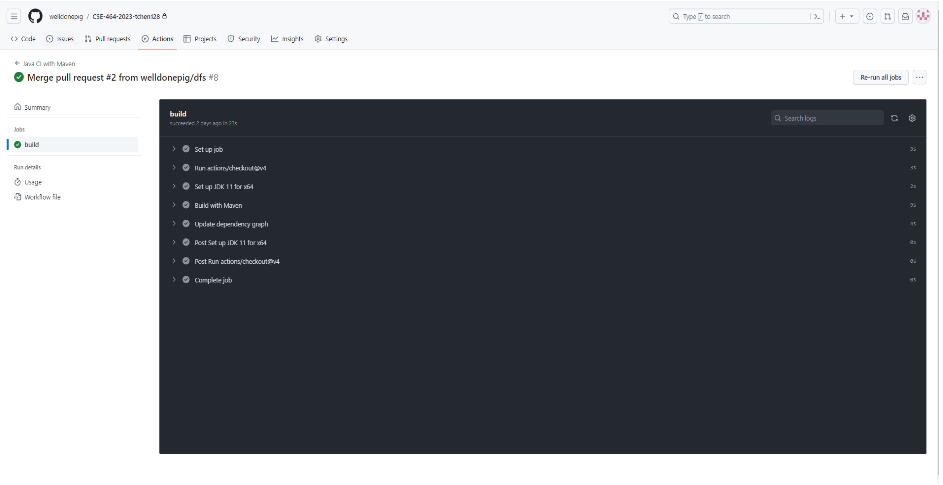
```
GraphParser parser = new GraphParser();
parser.addNode( label: "A");
parser.addNode( label: "B");
parser.addEdge( srcLabel: "A",  dstLabel: "B");
parser.removeEdge( srcLabel: "A",  dstLabel: "B");
System.out.println(parser.getEdges());
System.out.println(parser.getGraph());
```

Output:



```
Edge (A -> B) added.
Edge (A -> B) removed.
[]
([A, B], [])
```

2. Link and picture for GitHub Action intergrated into repository



Java CI with Maven

Merge pull request #2 from welldonepig/dfs #8

build
succeeded 2 days ago in 23s

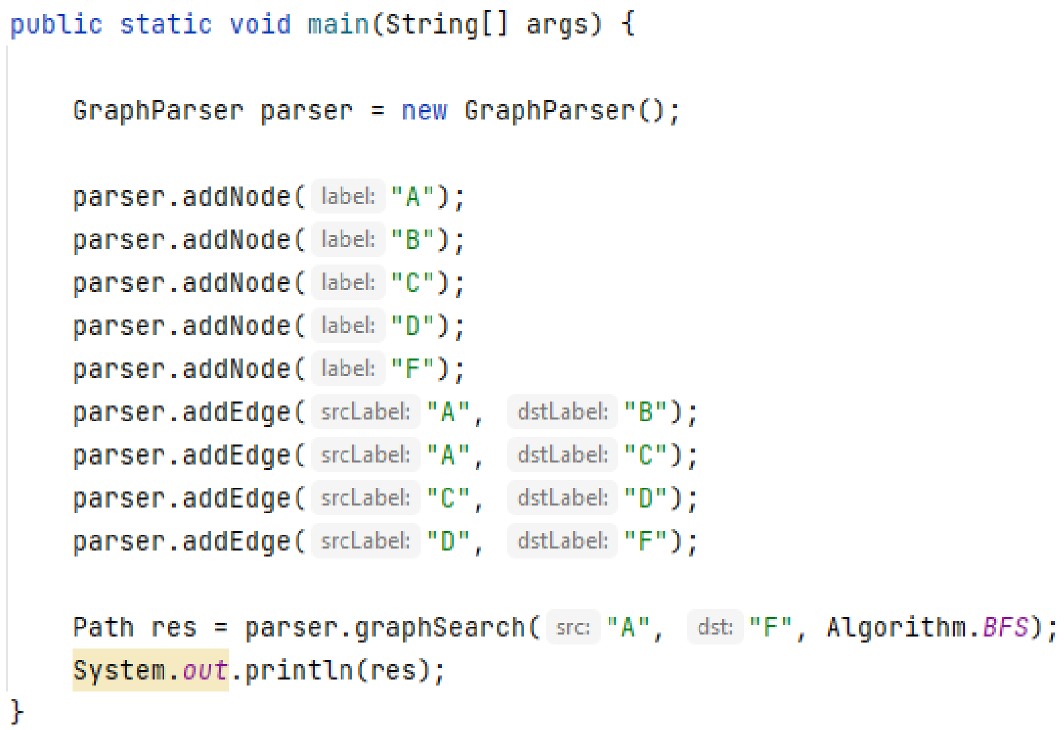
```
Set up job                     3s
Run actions/checkout@v4        3s
Set up JDK 11 for x64          2s
Build with Maven               9s
Update dependency graph        4s
Post Set up JDK 11 for x64     0s
Post Run actions/checkout@v4   0s
Complete job                   0s
```

Link Github Action: https://github.com/welldonepig/CSE-464-2023-tchen128/actions/runs/6743321937

Link commit: https://github.com/welldonepig/CSE-464-2023-tchen128/commit/711db0c051950d74eec4acfedb8b2f555e7ffd89

3. BFS

Link PR: https://github.com/welldonepig/CSE-464-2023-tchen128/pull/1
Example input:



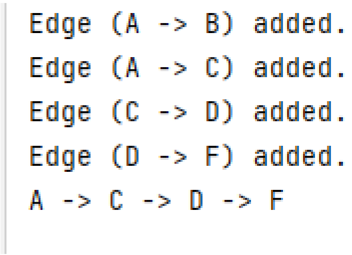
```
public static void main(String[] args) {

    GraphParser parser = new GraphParser();

    parser.addNode( label: "A");
    parser.addNode( label: "B");
    parser.addNode( label: "C");
    parser.addNode( label: "D");
    parser.addNode( label: "F");
    parser.addEdge( srcLabel: "A",  dstLabel: "B");
    parser.addEdge( srcLabel: "A",  dstLabel: "C");
    parser.addEdge( srcLabel: "C",  dstLabel: "D");
    parser.addEdge( srcLabel: "D",  dstLabel: "F");

    Path res = parser.graphSearch( src: "A",  dst: "F", Algorithm.BFS);
    System.out.println(res);
}
```

Output:

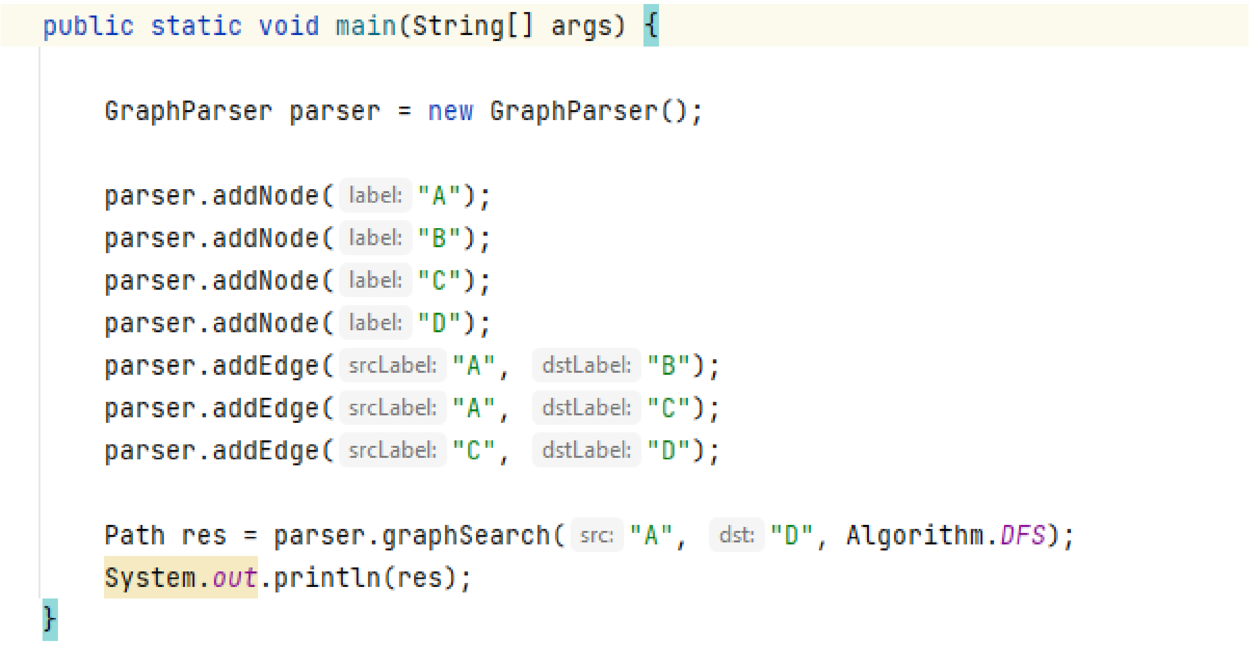


```
Edge (A -> B) added.
Edge (A -> C) added.
Edge (C -> D) added.
Edge (D -> F) added.
A -> C -> D -> F
```

4. DFS

Link PR: https://github.com/welldonepig/CSE-464-2023-tchen128/pull/2

Example input:



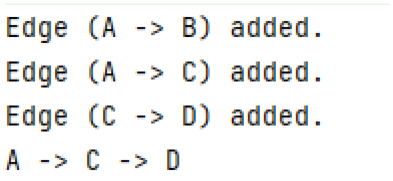
```java
public static void main(String[] args) {

    GraphParser parser = new GraphParser();

    parser.addNode( label: "A");
    parser.addNode( label: "B");
    parser.addNode( label: "C");
    parser.addNode( label: "D");
    parser.addEdge( srcLabel: "A",  dstLabel: "B");
    parser.addEdge( srcLabel: "A",  dstLabel: "C");
    parser.addEdge( srcLabel: "C",  dstLabel: "D");

    Path res = parser.graphSearch( src: "A",  dst: "D", Algorithm.DFS);
    System.out.println(res);
}
```

Output:



```
Edge (A -> B) added.
Edge (A -> C) added.
Edge (C -> D) added.
A -> C -> D
```

5. Merge bfs then merge dfs branch change search api with enum Algorithm

Link resolve conflict and merge commit: https://github.com/welldonepig/CSE-464-2023-tchen128/commit/d22dbbcc08879a94d73d11deb318d0776e9d846b