

**Rahul Bhiwande**  
**Program Structures & Algorithms**  
**Spring 2021**  
**Assignment No. 3**

- **Task –**

- Step 1:
  - (a) Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF\_HWQUPC. All you have to do is to fill in the sections marked with `// TO BE IMPLEMENTED ... // ...END IMPLEMENTATION`.
  - (b) Check that the unit tests for this class all work. You must show "green" test results in your submission (screenshot is OK).
- Step 2:

Using your implementation of UF\_HWQUPC, develop a UF ("union-find") client that takes an integer value  $n$  from the command line to determine the number of "sites." Then generates random pairs of integers between 0 and  $n-1$ , calling `connected()` to determine if they are connected and `union()` if not. Loop until all sites are connected then print the number of connections generated. Package your program as a static method `count()` that takes  $n$  as the argument and returns the number of connections; and a `main()` that takes  $n$  from the command line, calls `count()` and prints the returned value. If you prefer, you can create a main program that doesn't require any input and runs the experiment for a fixed set of  $n$  values. Show evidence of your run(s).
- Step 3:

Determine the relationship between the number of objects ( $n$ ) and the number of pairs ( $m$ ) generated to accomplish this (i.e., to reduce the number of components from  $n$  to 1). Justify your conclusion.
- **Relationship Conclusion:**

$m = c * n \cdot \ln(n)$ . (log to the base  $e$ )  
 $c \sim 1/2$

- Evidence to support the conclusion:

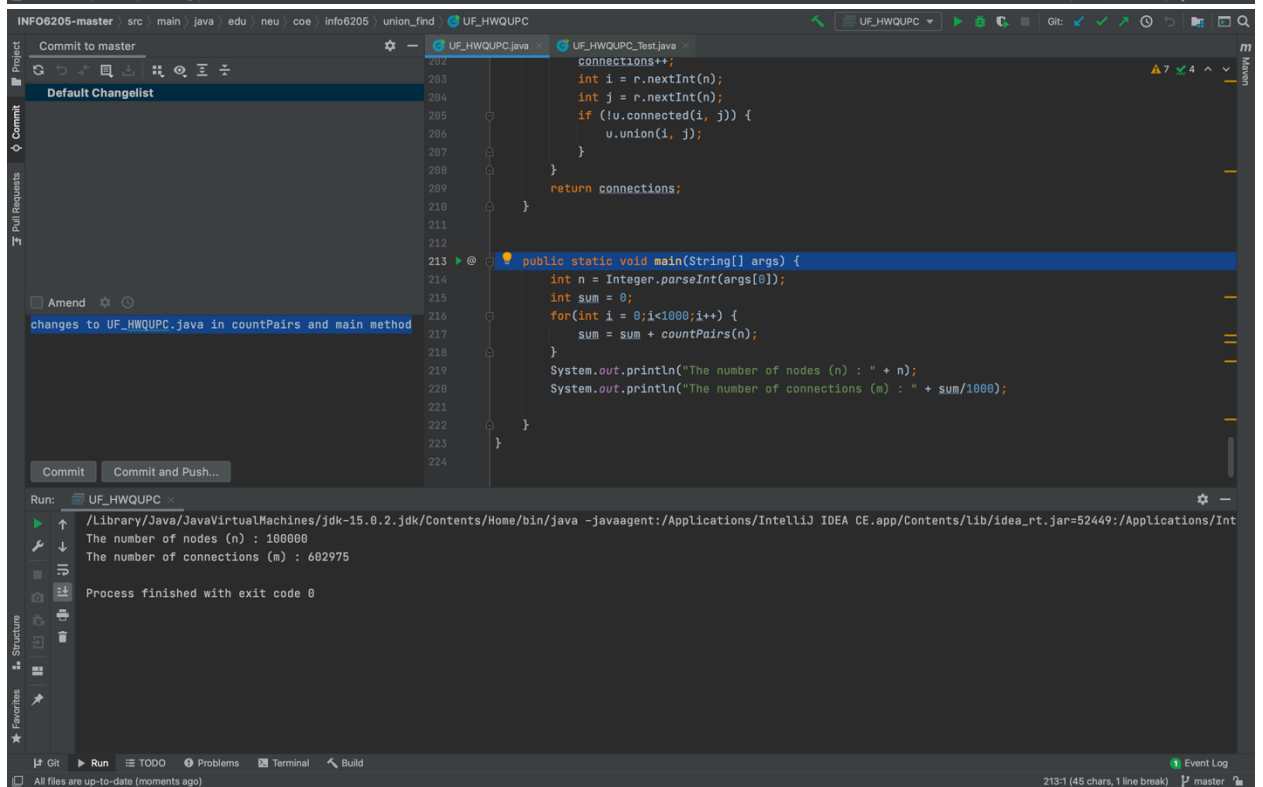
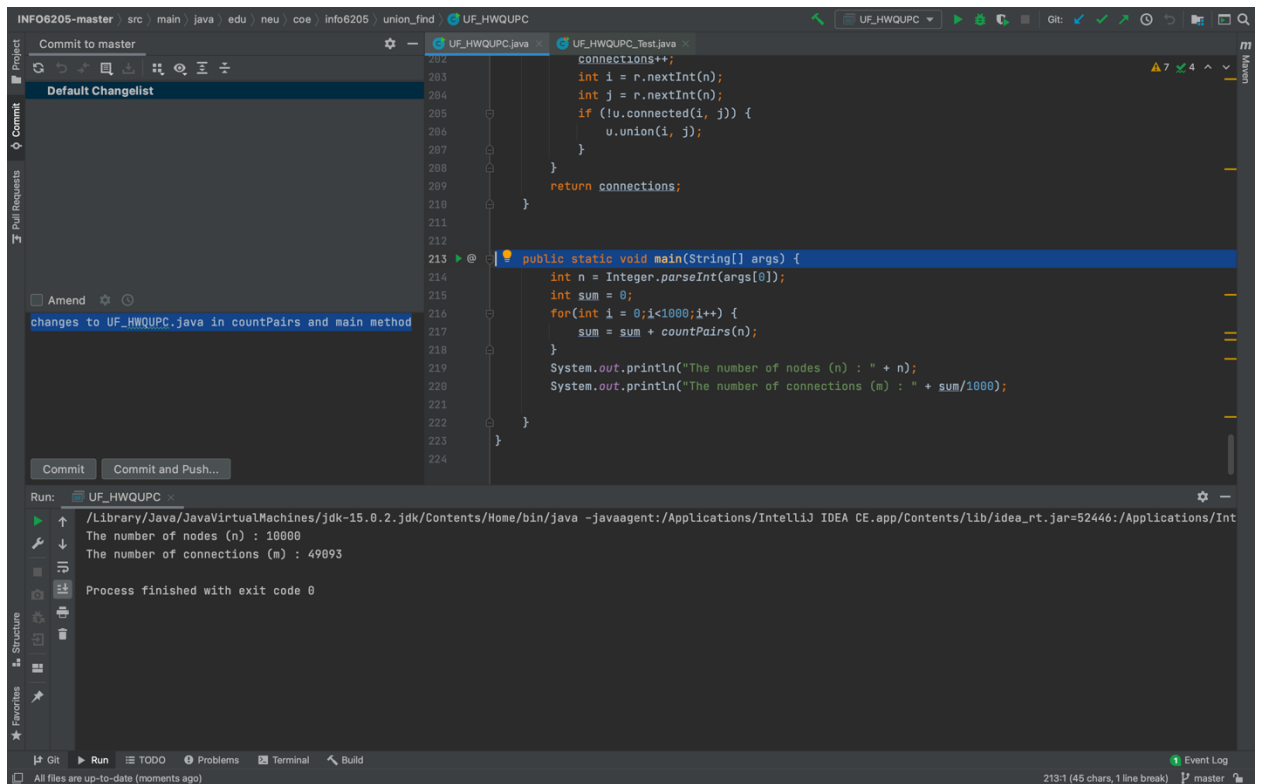
The image displays two screenshots of an IDE, likely IntelliJ IDEA, showing the process of testing a Java program. The top screenshot shows the code for `UF_HWQUPC.java` and `UF_HWQUPC_Test.java`. The code in `UF_HWQUPC.java` includes a `countPairs` method and a `main` method. The `main` method in `UF_HWQUPC_Test.java` calls `countPairs` with `n = 1000` and prints the results. The bottom screenshot shows the same code, but the `main` method in `UF_HWQUPC_Test.java` is modified to call `countPairs` with `n = 10000`. The output of the program is shown in the Run window, indicating that the number of nodes (n) is 1000 and the number of connections (m) is 262 in the top screenshot, and 10000 and 3770 in the bottom screenshot.

```
INFO6205-master | src | main | java | edu | neu | coe | info6205 | union_find | UF_HWQUPC | UF_HWQUPC_Test.java x
Commit to master
Default Changelist
Amend
changes to UF_HWQUPC.java in countPairs and main method
Commit
Commit and Push...

Run: UF_HWQUPC x
/Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52439:/Applications/Int
The number of nodes (n) : 100
The number of connections (m) : 262
Process finished with exit code 0

INFO6205-master | src | main | java | edu | neu | coe | info6205 | union_find | UF_HWQUPC | UF_HWQUPC_Test.java x
Commit to master
Default Changelist
Amend
changes to UF_HWQUPC.java in countPairs and main method
Commit
Commit and Push...

Run: UF_HWQUPC x
/Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52443:/Applications/Int
The number of nodes (n) : 10000
The number of connections (m) : 3770
Process finished with exit code 0
```



## • Unit tests result:

The screenshot displays an IDE interface with the following components:

- Project View:** Shows the project structure for 'INFO6205-master [INFO6205]'. The 'target' directory is expanded, showing files like 'gitignore', 'INFO6205.iml', 'LICENSE', 'pom.xml', and 'README.md'.
- Code Editor:** Displays the source code for 'UF\_HWQUPC\_Test.java'. The code includes a package declaration, imports, and a public class 'UF\_HWQUPC\_Test' with a test method 'testToString()'.
- Run View:** Shows the execution results of the tests. The status bar indicates 'Tests passed: 13 of 13 tests - 12ms'.
- Test Results Table:** A table listing the individual tests and their execution times.

Test Name	Duration
testIsConnected01	3ms
testIsConnected02	1ms
testIsConnected03	5ms
testFind0	0ms
testFind1	0ms
testFind2	0ms
testFind3	1ms
testFind4	2ms
testFind5	0ms
testToString	0ms
testConnect01	0ms
testConnect02	0ms
testConnected01	0ms

The bottom status bar shows 'Tests passed: 13 (a minute ago)', '4:1', 'UTF-8', '4 spaces', and 'master'.