Program Structures & Algorithms

Spring 2022

Assignment No. 3

Name: Yuan Huang

(NUID): 002117054

- **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 in terms of your observations and what you think might be going on.

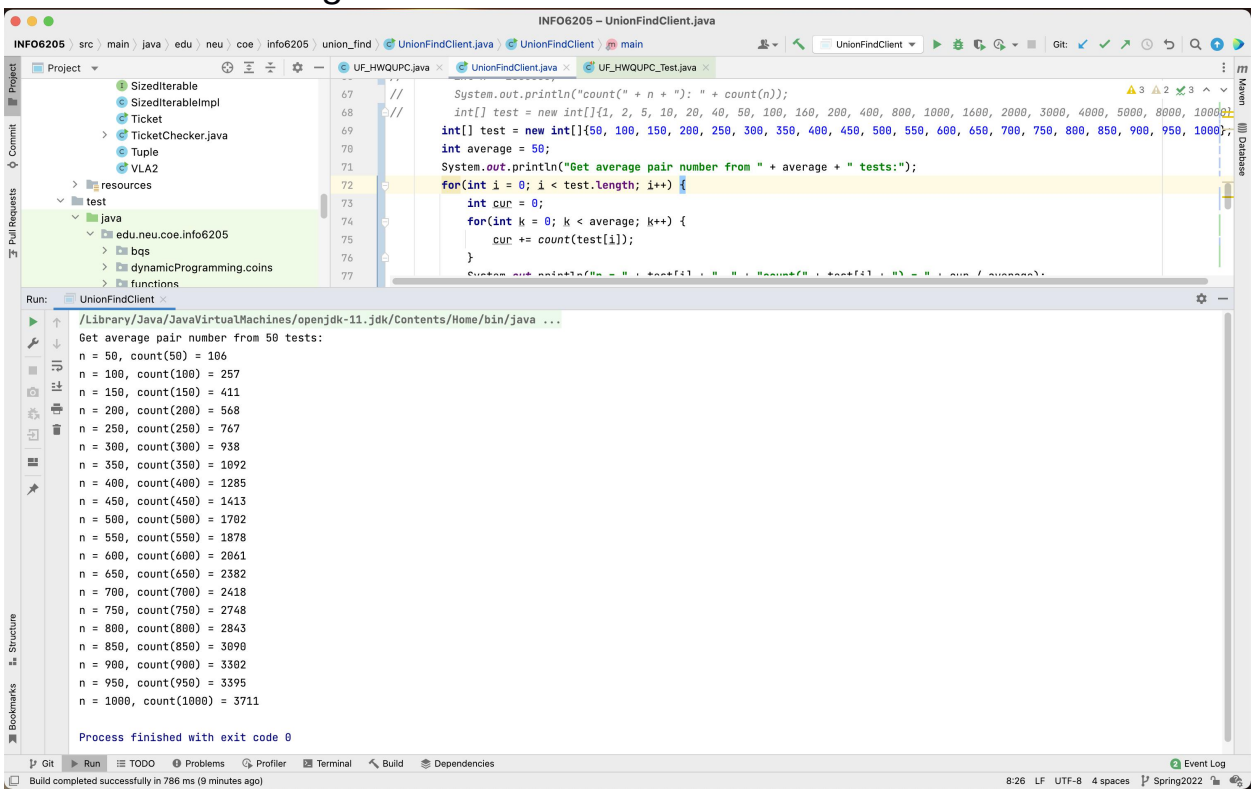
NOTE: although I'm not going to tell you in advance what the relationship is, I can assure you that it is a simple relationship.

- **Output screenshot**

Part2 output:

I created a UnionFindClient class for part 2 & 3. It has a static function called count(), which is used for “m = count(n)”. Input number of objects n, and output the number of pairs m generated to union all of n objects to only one component. In the main function it is the testing process. There is a predefined array named test[], which contains a list of different input n range from 50 to 1000.

Each m is an average result of 50 times tests.

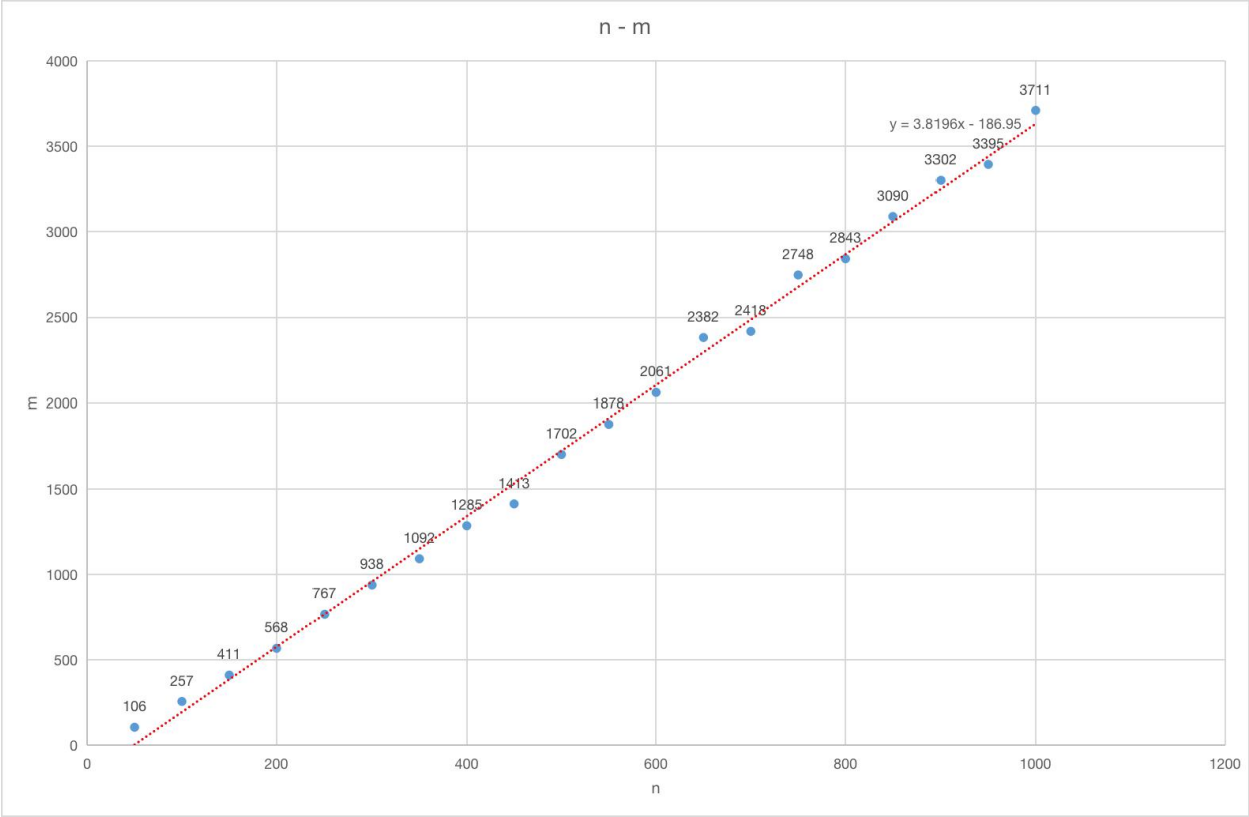


- **Relationship Conclusion**

$$m = 3.8196 * n - 186.95$$

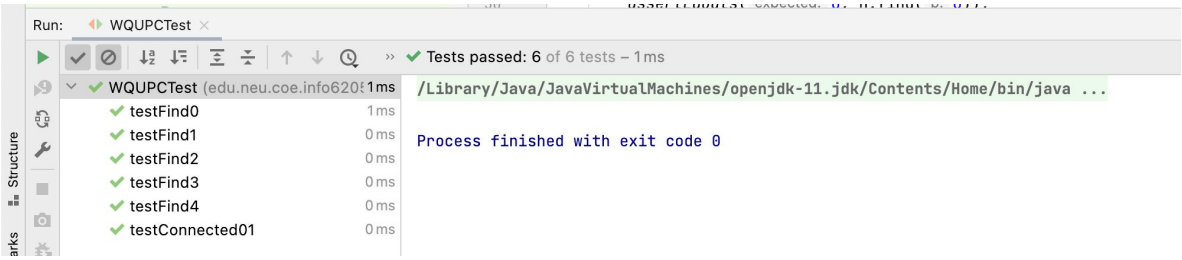
- **Evidence / Graph**

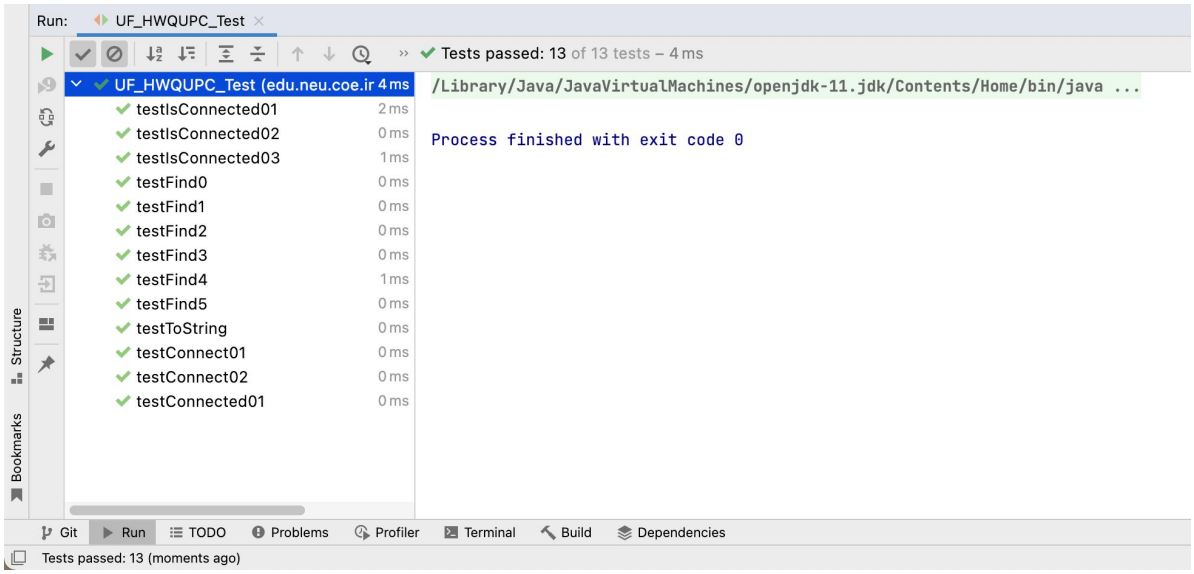
The blue dots are test result.
The red line is the relationship conclusion.



- **Unit tests result**

Test result:




Run:
UF_HWQUPC_Test
Tests passed: 13 of 13 tests – 4 ms
UF_HWQUPC_Test (edu.neu.coe.ir 4 ms
testIsConnected01 2 ms
testIsConnected02 0 ms
testIsConnected03 1 ms
testFind0 0 ms
testFind1 0 ms
testFind2 0 ms
testFind3 0 ms
testFind4 1 ms
testFind5 0 ms
testToString 0 ms
testConnect01 0 ms
testConnect02 0 ms
testConnected01 0 ms
/Library/Java/JavaVirtualMachines/openjdk-11.jdk/Contents/Home/bin/java ...
Process finished with exit code 0
Structure
Bookmarks
Git
Run
TODO
Problems
Profiler
Terminal
Build
Dependencies
Tests passed: 13 (moments ago)