**Program Structures & Algorithms**

**Spring 2022**

**Assignment No. N3**

Name: Chengyan Mu

(NUID): 002198861

* **Task**
* **Output screenshot**
* **Relationship Conclusion**
* **Evidence / Graph**
* **Unit tests result**
* **Task**

**Step 1:**

1. **.**  Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF\_HWQUPC.

(b)Check that the unit tests for this class all work. You must show "green" test results in your submission

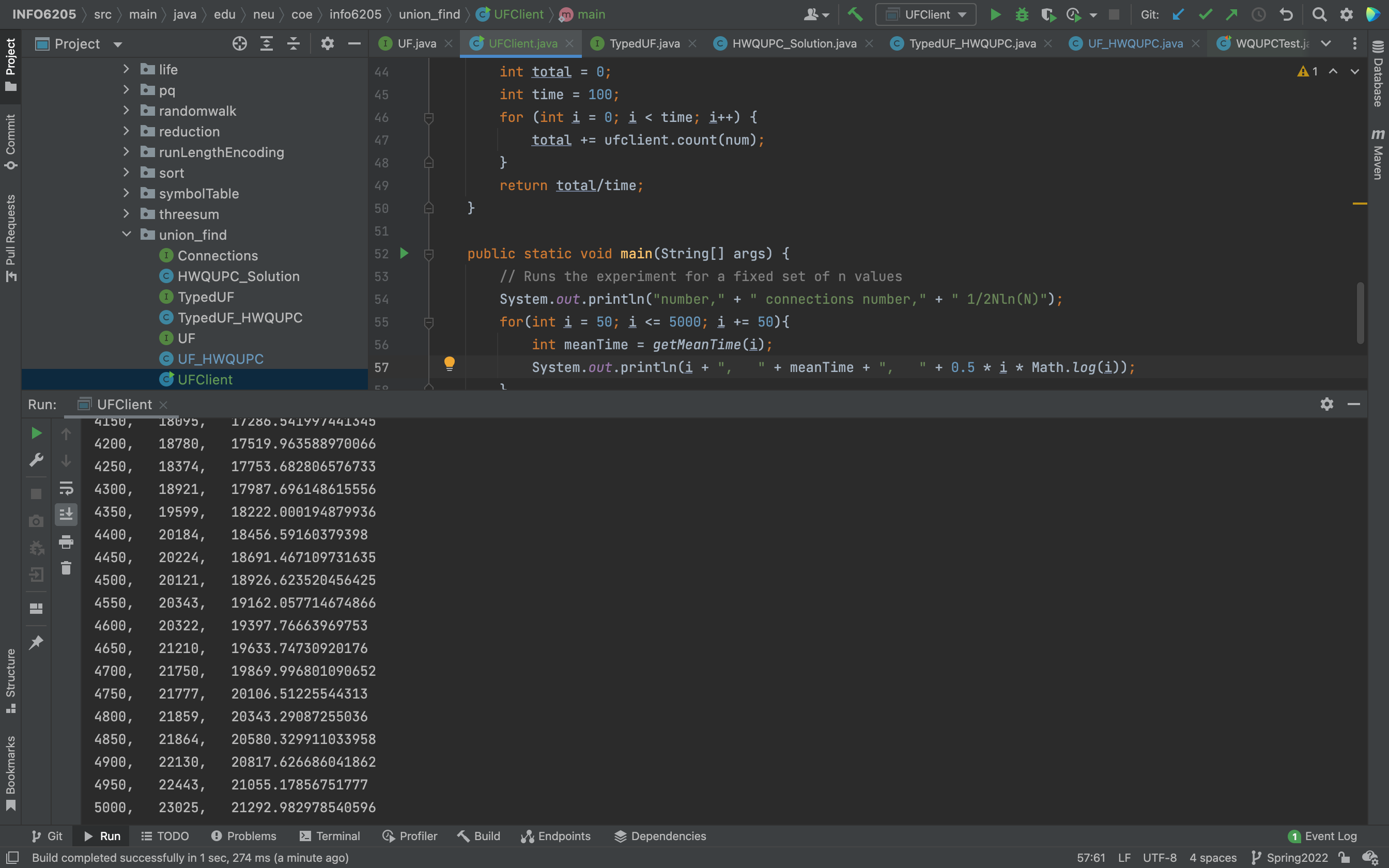
**Step 2:**

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).

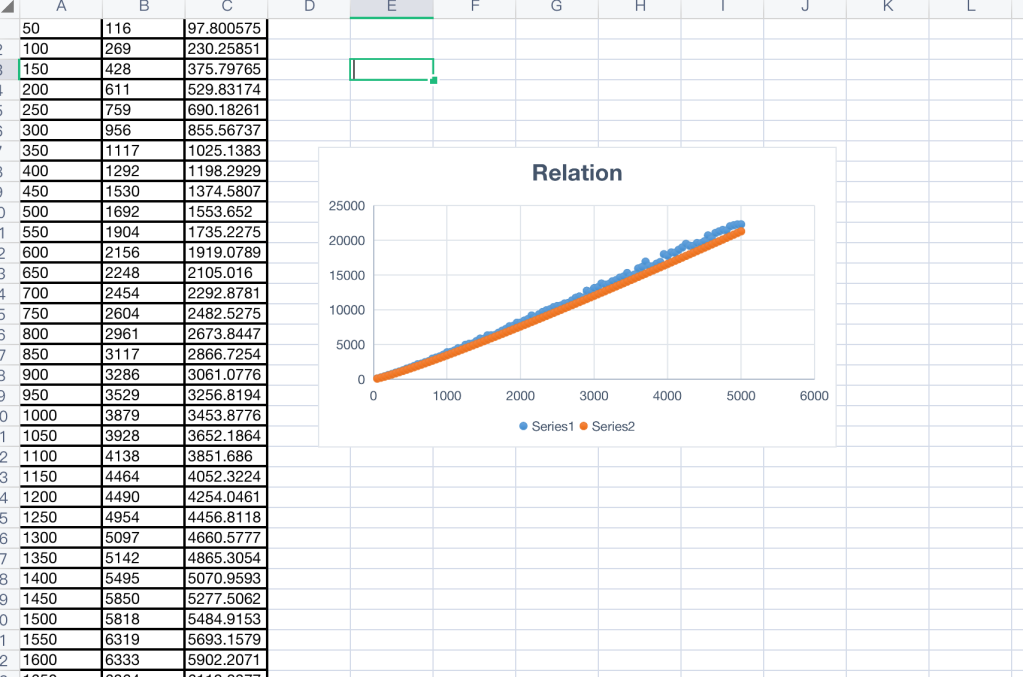
**Step 3:**

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.

* **Output screenshot**



* **Relationship Conclusion**
* **Evidence / Graph**

****

* **Unit tests result**

