**Program Structures & Algorithms**

**Spring 2022**

**Assignment No. 3**

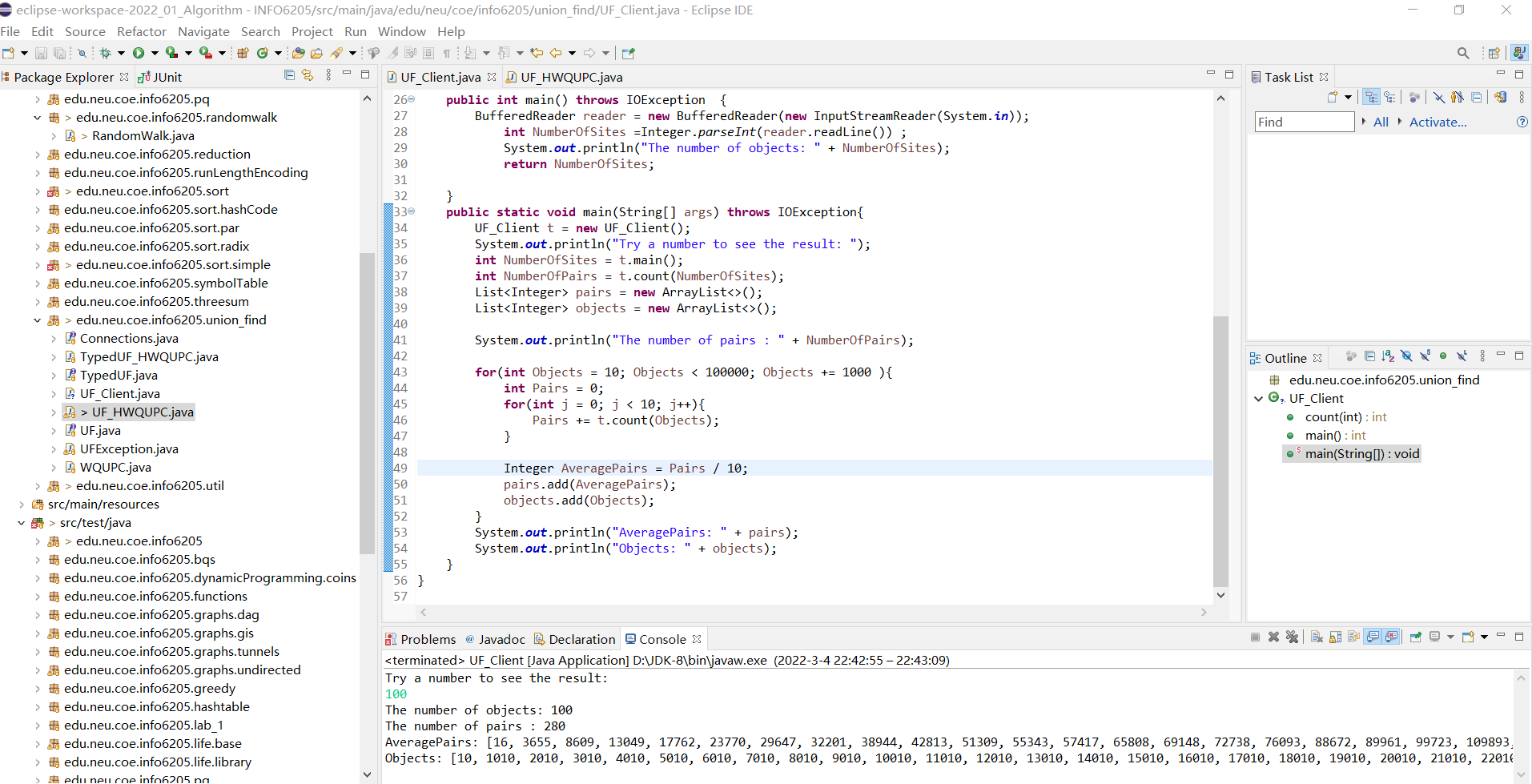
Name: Xinlin Ying

(NUID): 001535622

* **Task**
* Your task is
* Step 1:  
  (a) Implement height-weighted Quick Union with Path Compression.
* (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.
* **Output screenshot**

UF\_Client.java

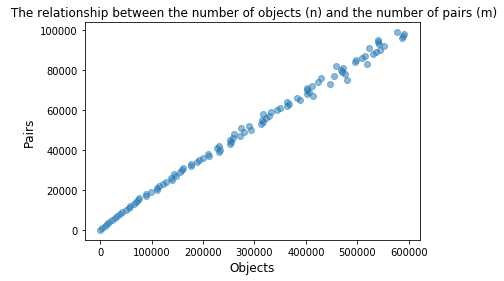
Run the static main(String[] args) to make user input a number to test the method and get the result of 1000000 experiments.



* **Relationship Conclusion**

It’s similar to a linear relationship.

* **Evidence / Graph**

****

Part of the data:

|  |  |
| --- | --- |
| **Number of Objects(n)** | **Number of Pairs(m)** |
| 1010 | 3655 |
| 7010 | 32201 |
| 15010 | 69148 |
| 22010 | 110854 |
| 28010 | 137132 |
| 37010 | 188643 |
| 41010 | 231290 |
| 51010 | 274450 |
| 65010 | 312203 |
| 76010 | 382776 |
| 86010 | 471721 |
| 99010 | 577352 |

* **Unit tests result**

