Peng Chen (001098655)

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No.3**

GitHub: <https://github.com/pngchen/INFO6205-Assignments/tree/main/assignment3>

**Task:**

* (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).

* 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).
* 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.

**Conclusion:**

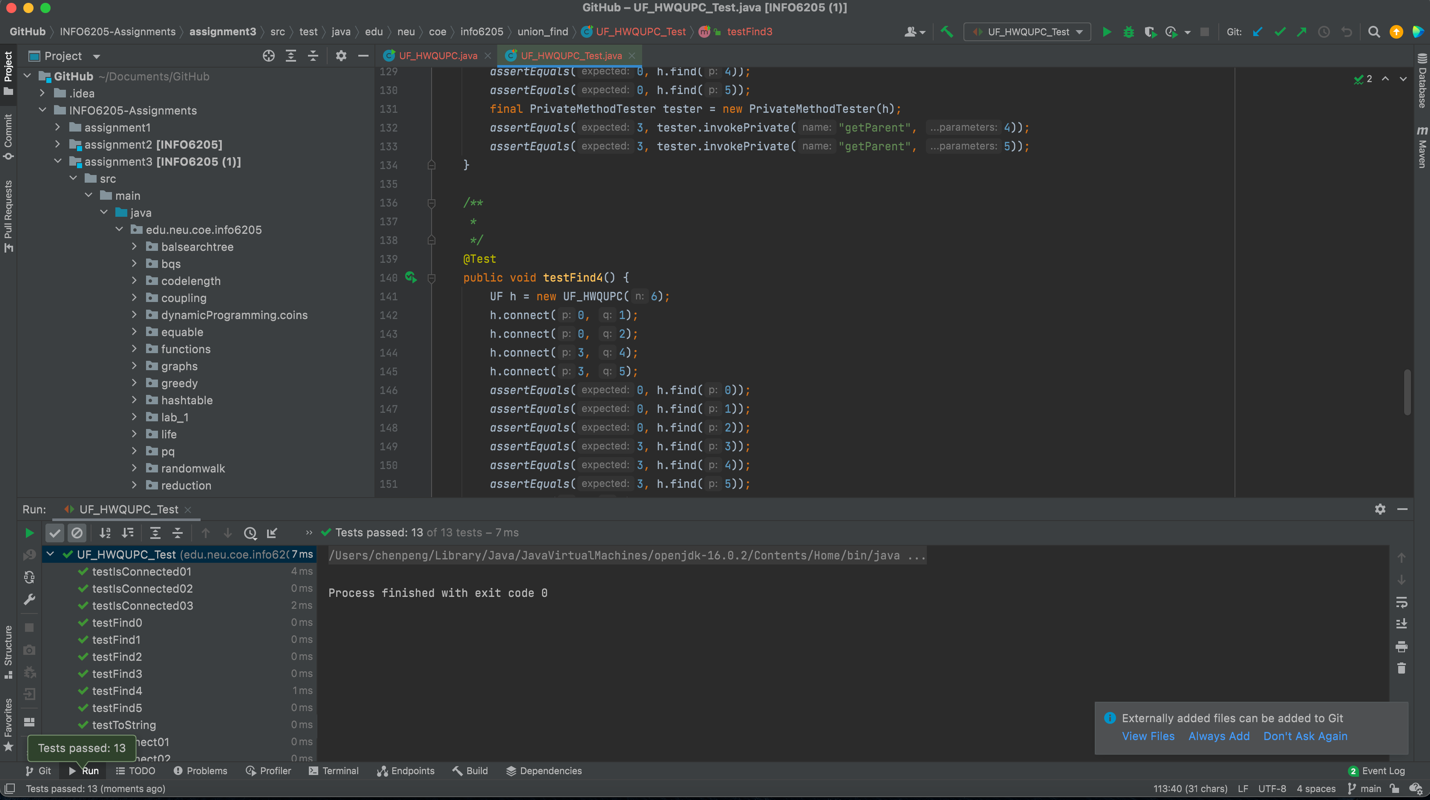
The relationship between the number of objects (n) and the number of pairs (m):

You can see in the picture below. This figure is about the relationship between a and pairs. You can see more details in my Excel file.

After several tests, I found the pair(m) is directly proportional to nlogn, and the average slope is 0.38. you also can see more details about this in the Excel file.

**Evidence:**

Part1:



Part2:

