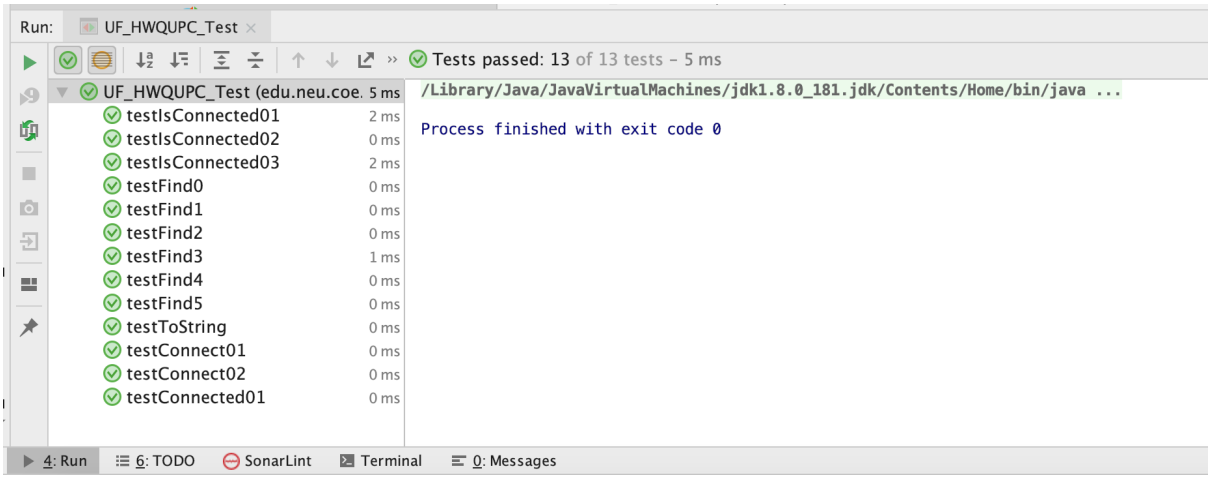
Charmi Dalal (001582441)

INFO 6205 Program Structure & Algorithms
Spring 2021
Assignment 3

Task 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).

Output:

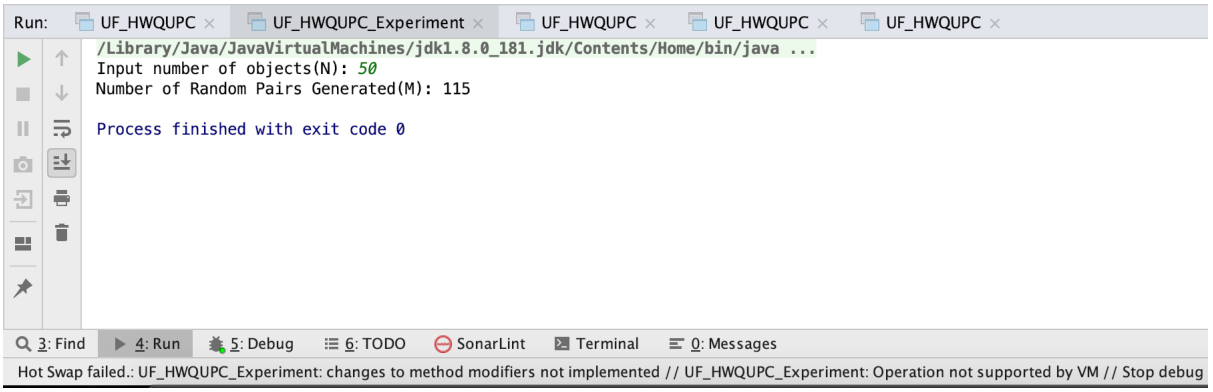


1.1 Test Case Results

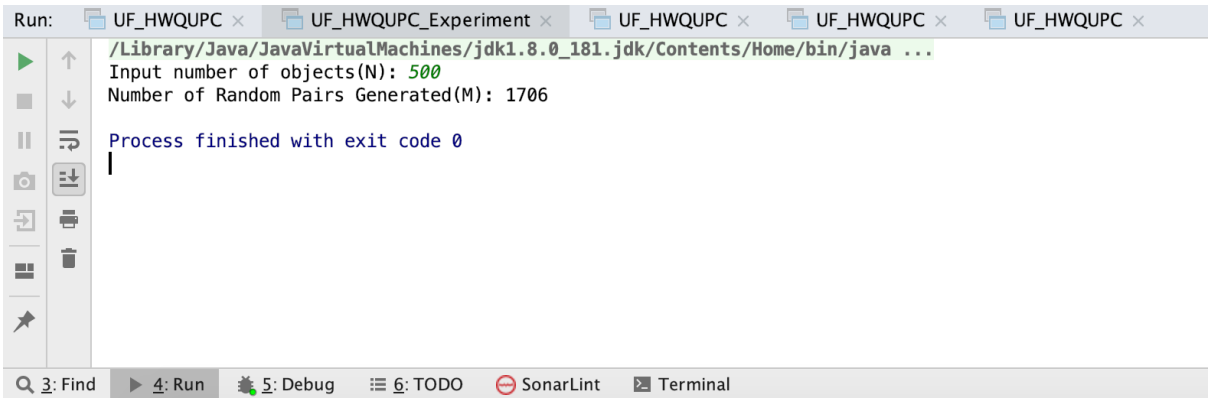
Task 2:

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).

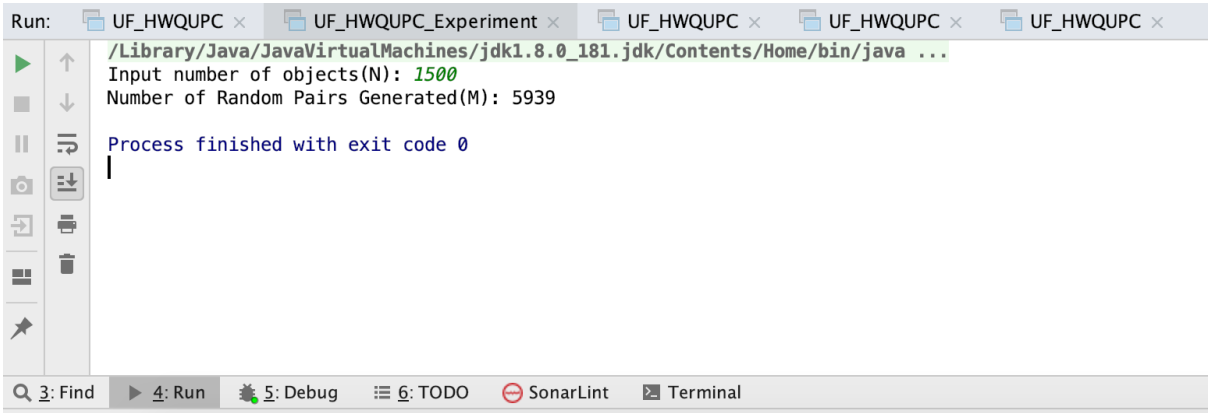
Output:



2.1 Run Result



2.2 Run Result



2.3 Run Result

Task 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.

Don't forget to follow the submission guidelines. And to use sufficient (and sufficiently large) different values of n.

Relationship Conclusion:

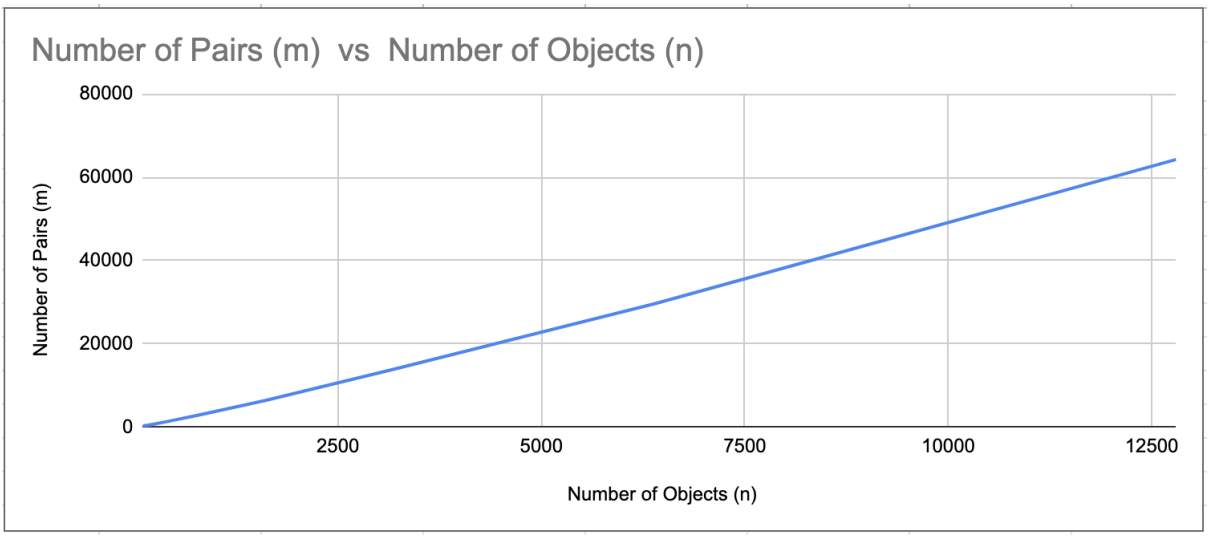
From the multiple experiments I found that the expected number of pairs (m) is very close to **n* ln(n)* 0.5** where n is the total number of elements entered as input. Below are the results of the number of experiments I have done and we can clearly see that the ratio of number of actual pairs and number of expected pairs from our equation is near to 1 which proves my conclusion.

m ≈ n* ln(n)* 0.5

Equation: n logn / 2			
Number of Objects (n)	**Number of Pairs (m)**	**Expected Pairs (n logn / 2)**	**Ratio**
100	262	230	1.138
200	589	530	1.112
400	1319	1198	1.101
800	2929	2674	1.095
1600	6345	5902	1.075
3200	13952	12913	1.080
6400	29656	28045	1.057
12800	64257	60526	1.062

3.1 Experiment Summary

In theory, Weight Quick-Union with Path Compression is not quite linear. But In practice, Weight Quick-Union with Path Compression is linear.



3.2 Linear Path Compression