Program Structures & Algorithms  
Spring 2022  
Assignment No. 3  
Name: SAITARUN MOVVA  
(NUID):002922800

**Task**

Step 1:

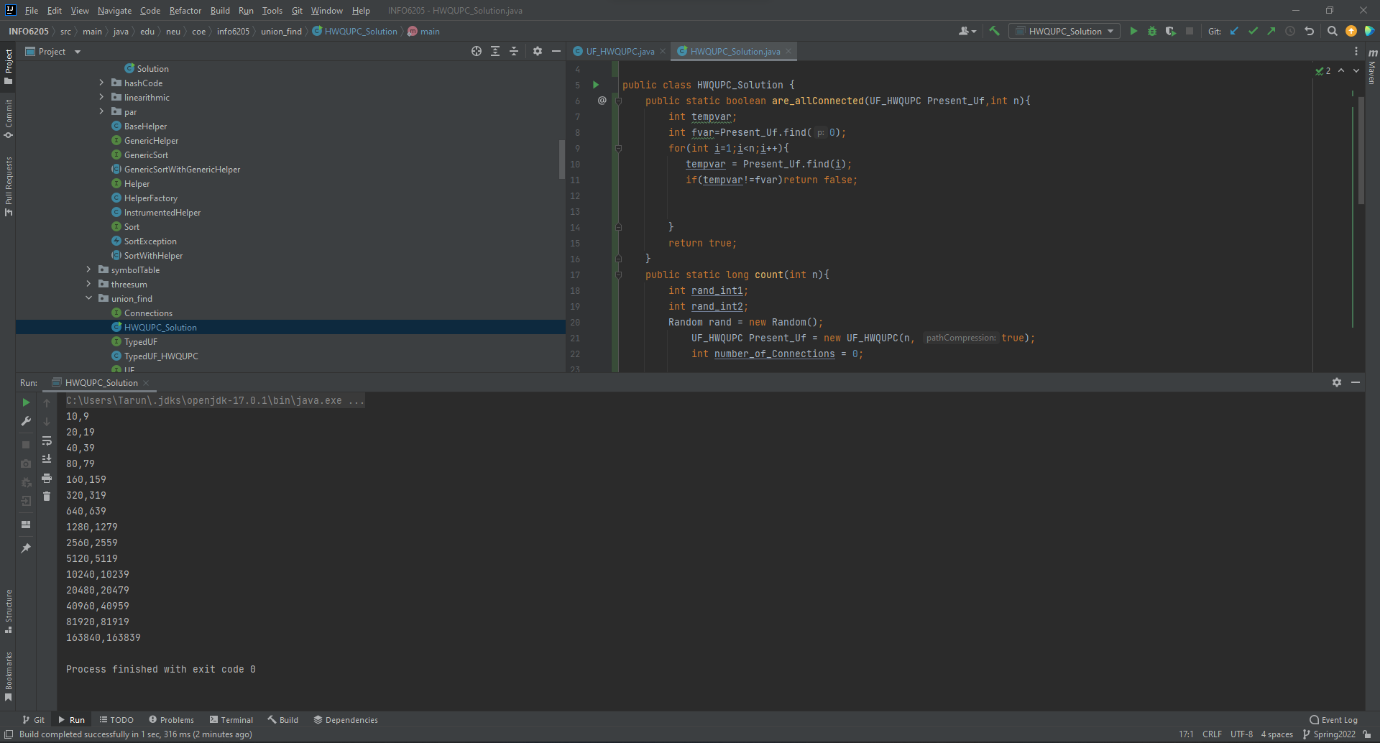
(a) To implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF\_HWQUPC. All you have to do is to fill in the sections marked with // TO BE IMPLEMENTED ... // ...END IMPLEMENTATION.

(b)To check that the unit tests for this class all work. You must show "green" test results in your submission (screenshot is OK).

Step 2:  
To use 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:  
To 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.

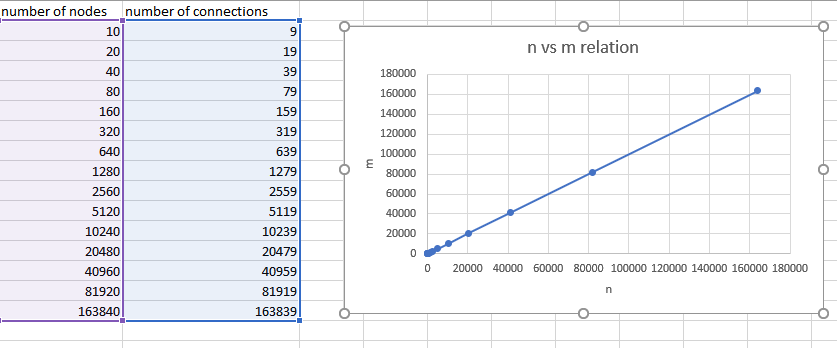
**Output screenshot**



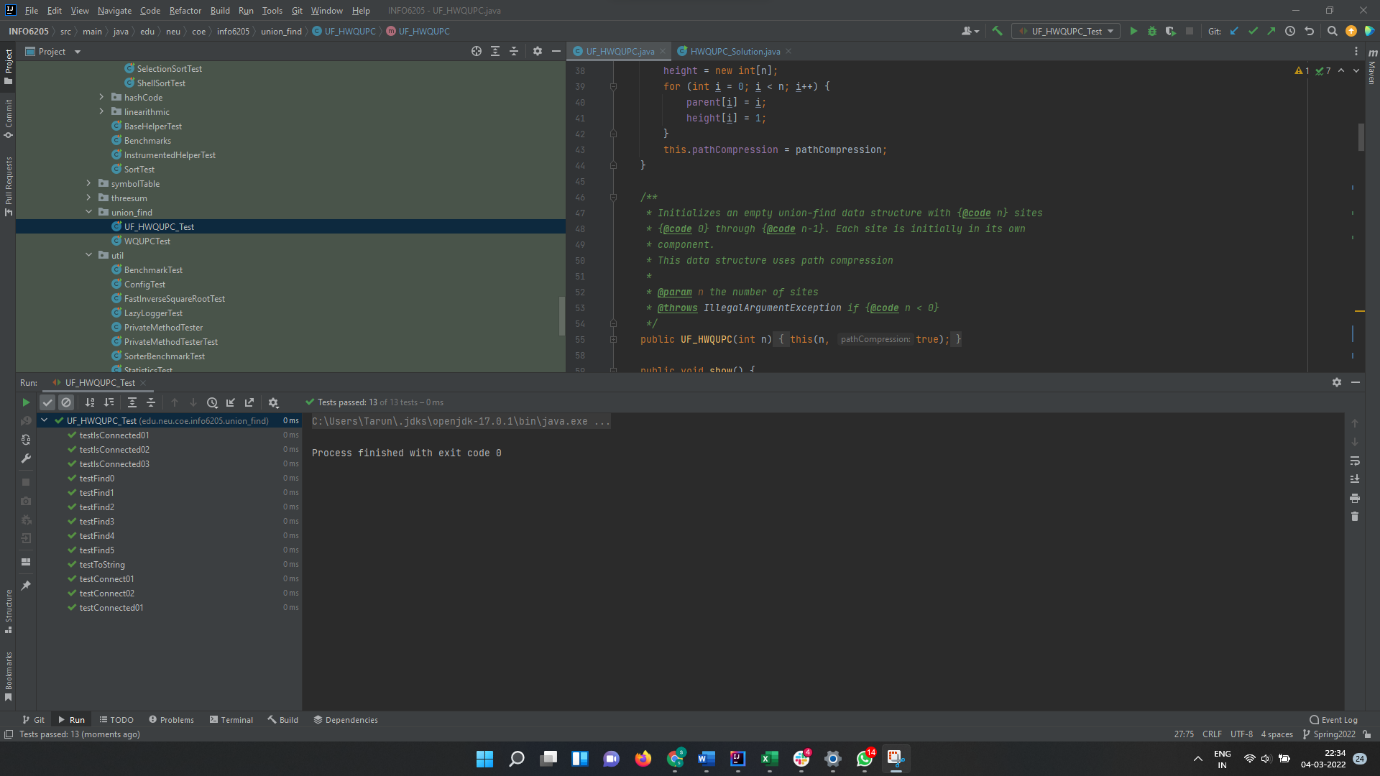
**Relationship Conclusion**

Here after generating values for number of objects(n) vs number of pairs/Connections in graph(m) or connections in the graph for them to be fully connected comes out to be m=n-1.

**Evidence / Graph**



**Unit Test Result**

****

Thanking you,

Saitarun Movva