Program Structures & Algorithms
Spring 2022

Assignment No - 3 (WQUPC)

Name: Naina Rajan
NUID: 002922398

Task -

Step 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).

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:

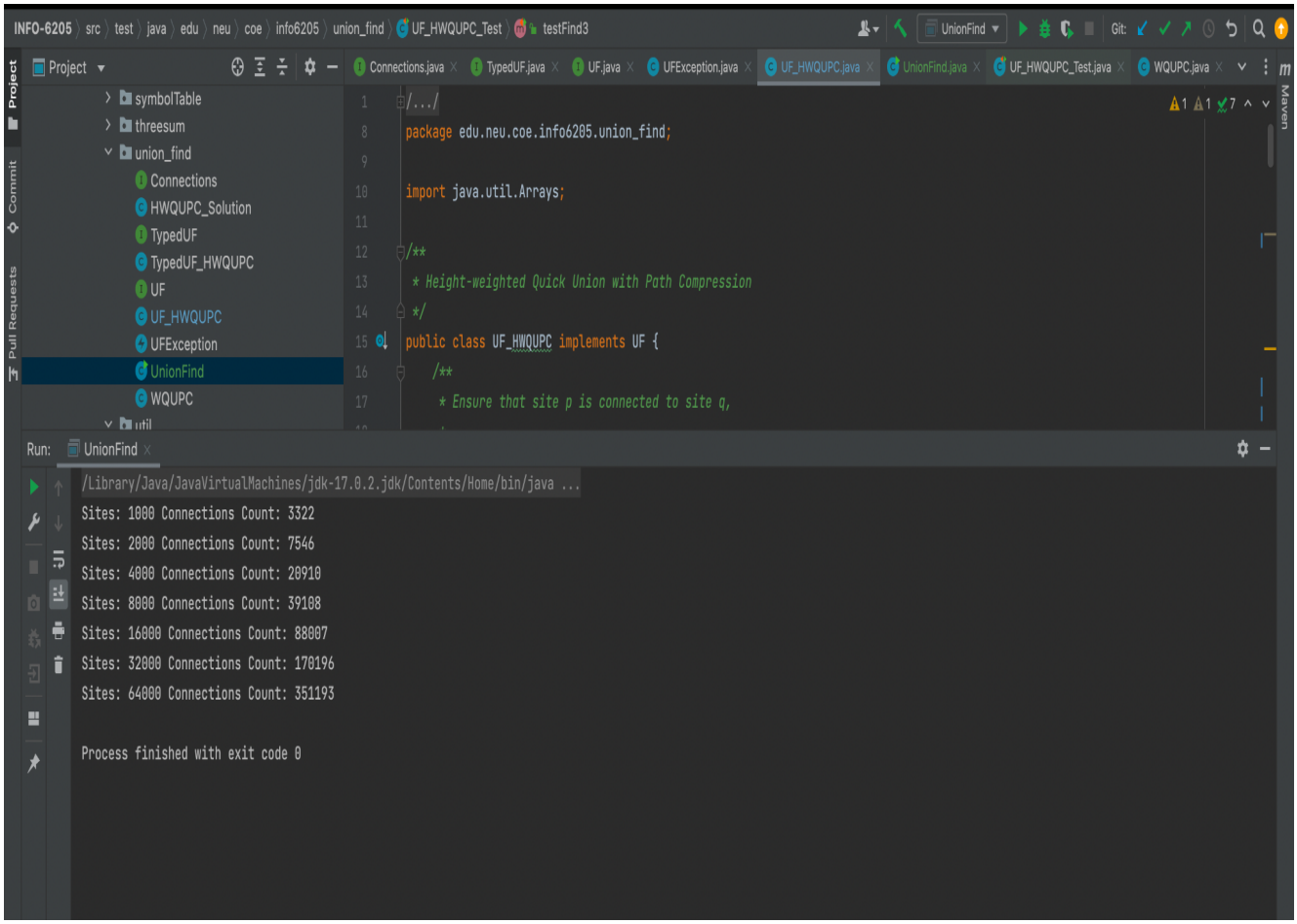
Determine the relationship between the number of objects (*n*) and the number of pairs (*m*) generated to accomplish this (i.e. to reduce the number of components from *n* to 1). Justify your conclusion in terms of your observations and what you think might be going on.

NOTE: although I'm not going to tell you in advance what the relationship is, I can assure you that it is a *simple* relationship.

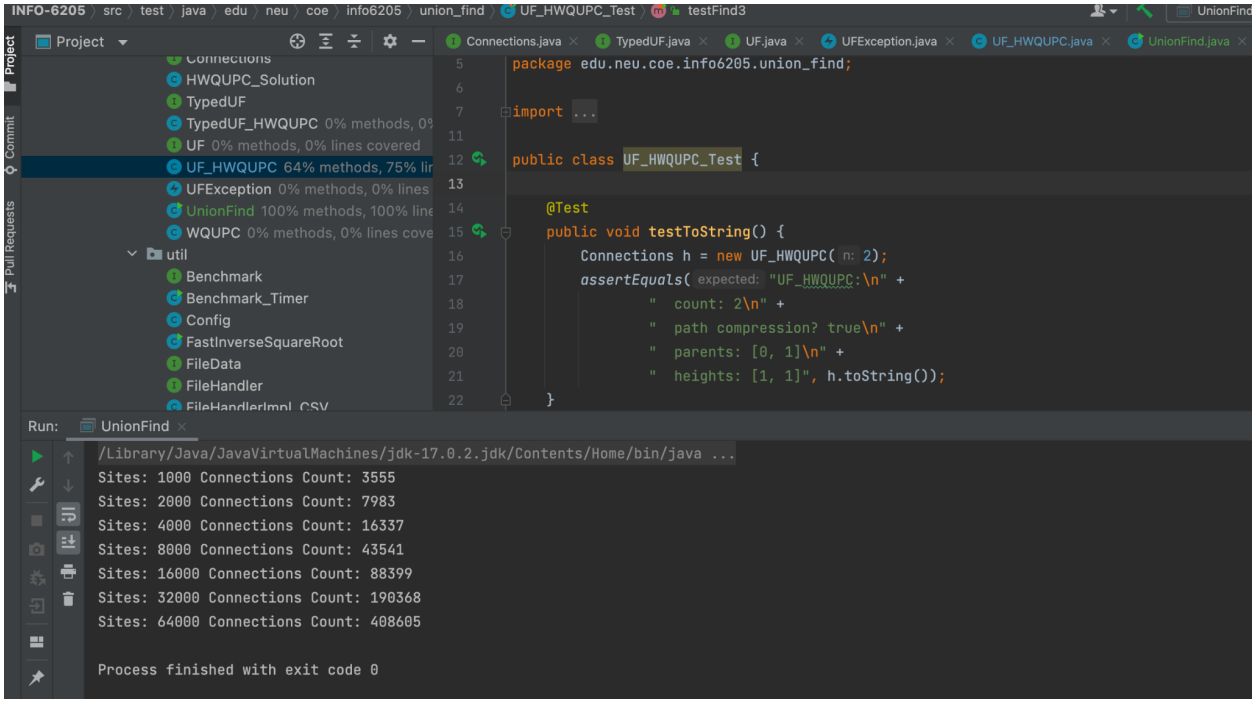
Don't forget to follow the submission guidelines. And to use sufficient (and sufficiently large) different values of n.

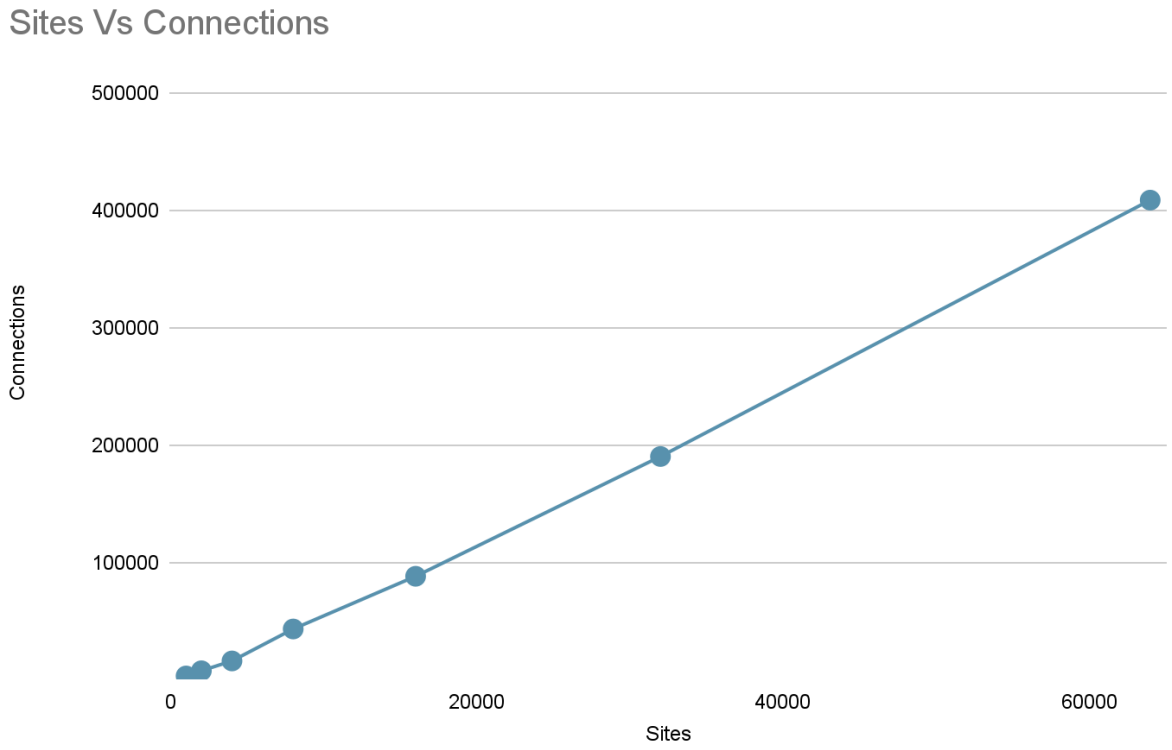
Output Screenshot

- Implemented the functions **mergeComponents(), doPathCompression()** and **find()**
- Successfully ran the code



Relationship Conclusion



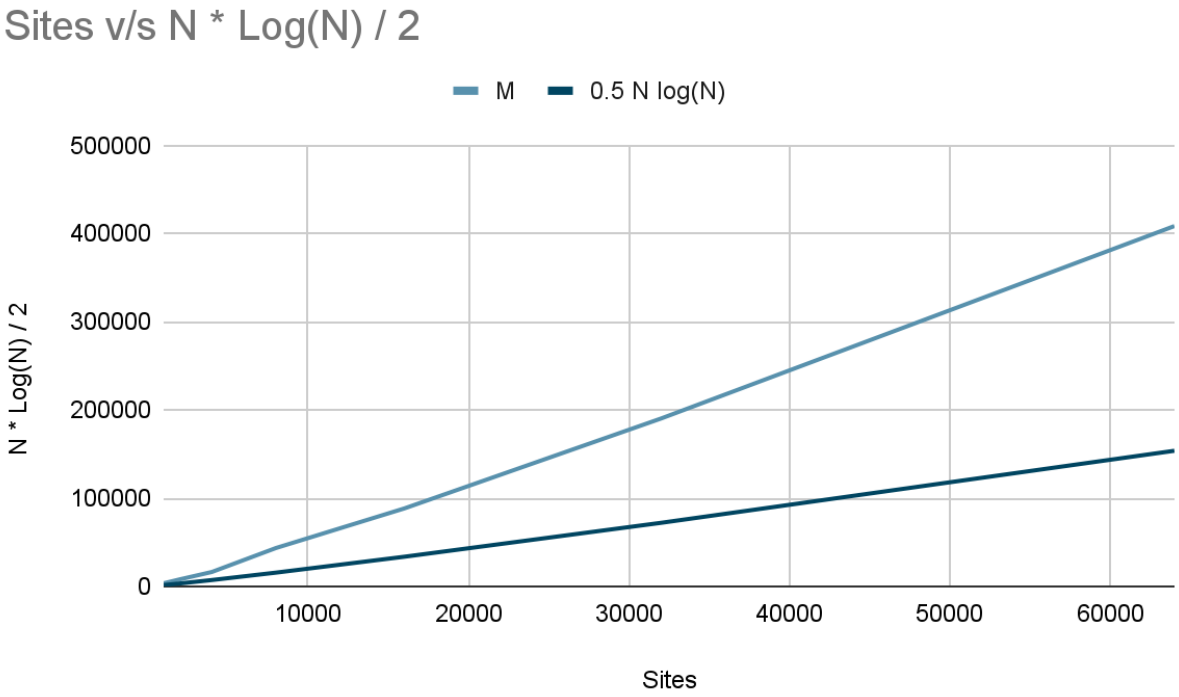


1. From the above Output Screenshot and Table, we can conclude:

   Relationship between n(number of sites) and m(number of connections)is: M = (N * log (N)) / 2

   The relationship is linear. As the number of sites grow, the number of connections grow linearly.

Evidence/Graph

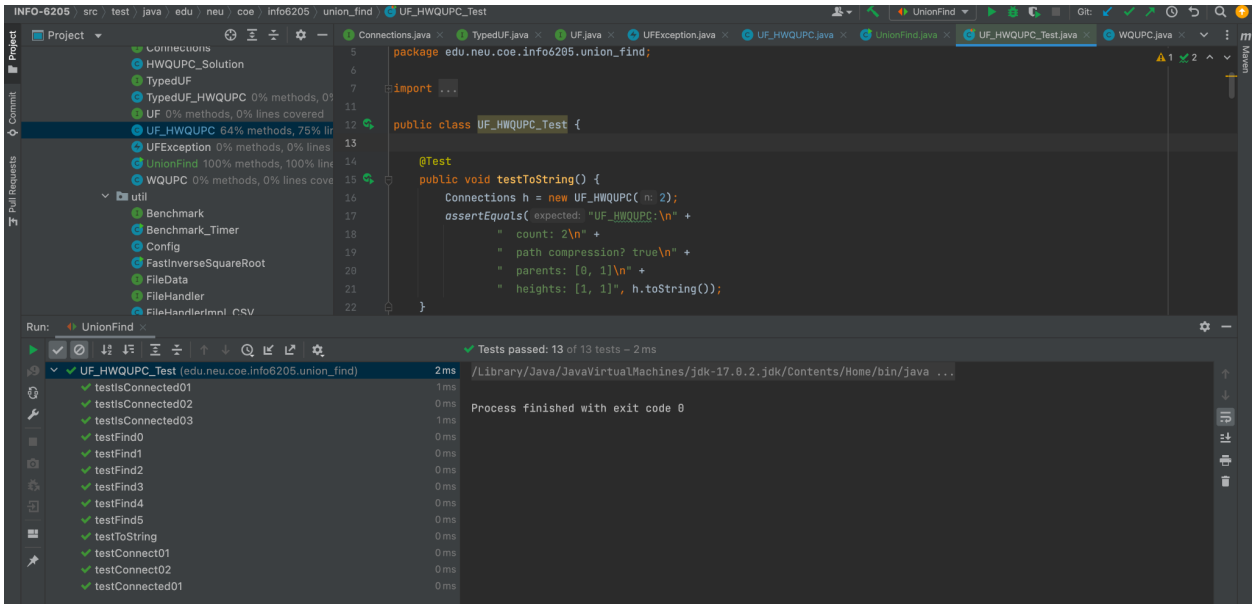


Hence the relation should be M = (N * Log(N)) / 2

Where M = Connection Count and N = Sites

Sites	M	0.5 N log(N)
1000	3555	1500
2000	7983	3301
4000	16337	7204
8000	43541	15612
16000	88399	33633
32000	190368	72083
64000	408605	153798

Unit Tests Result



Git Repository -

https://github.com/Naina-NEU/INFO6205/commit/930e76142b4ed7bfab481176257c6381cc0cd81f