

Program Structures and Algorithms
Spring 2023(SEC –01)
Assignment-4

NAME: Ashi Tyagi
NUID: 002706544

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.

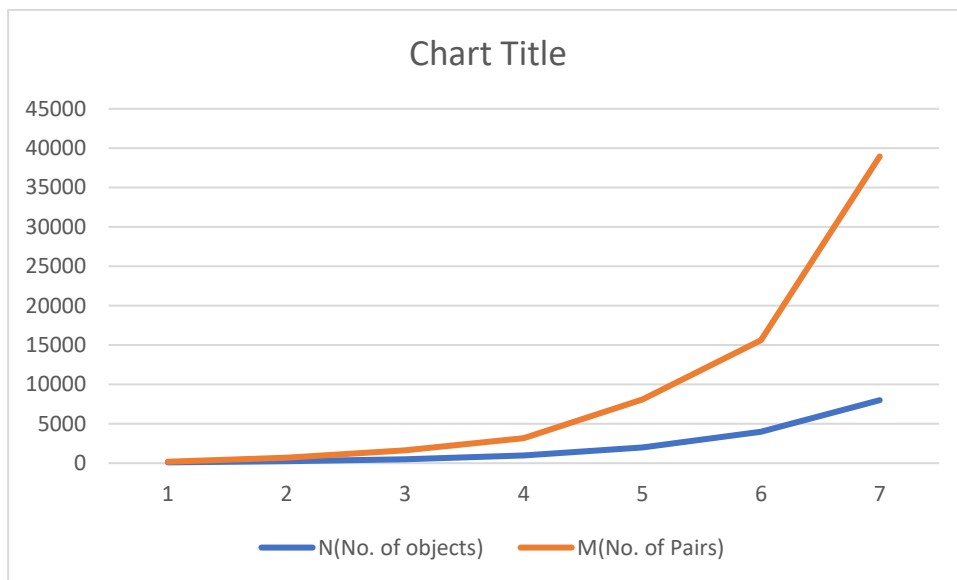
NOTE: although I'm not going to tell you in advance what the relationship is, I can assure you that it is a *simple* relationship.

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

Relationship Conclusion/ Evidence Graph:

N(No. of objects)	M(No. of Pairs)
----------------------	--------------------

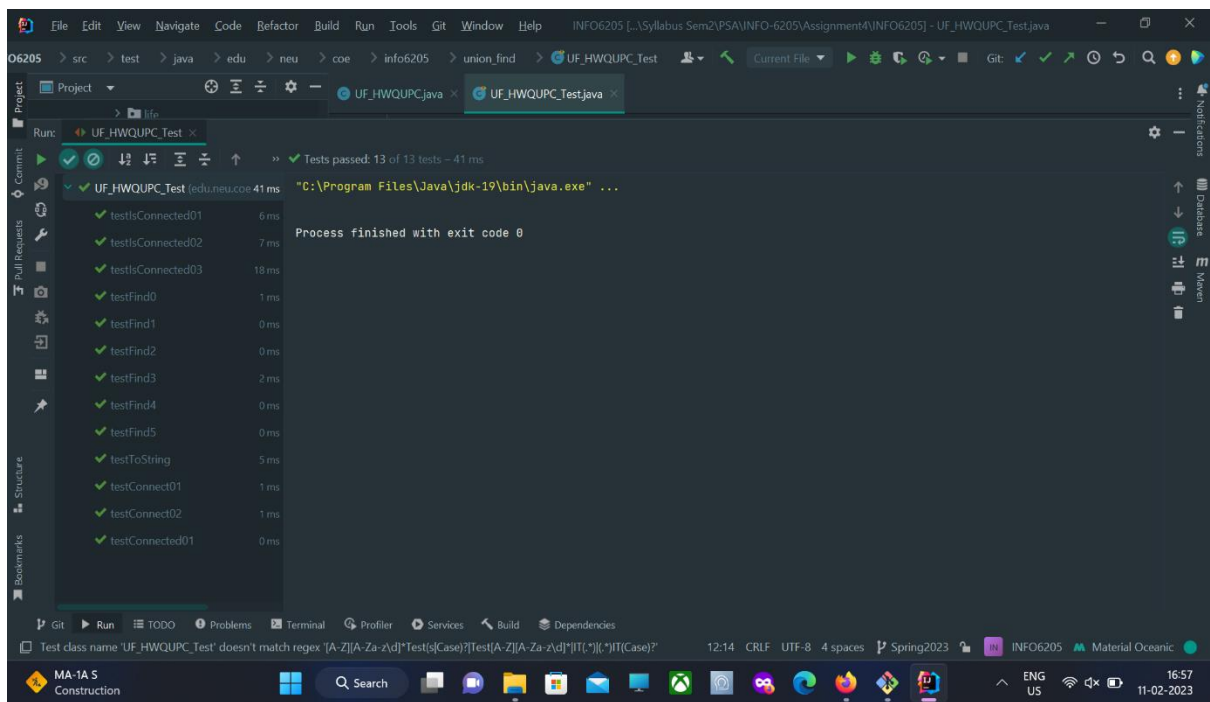
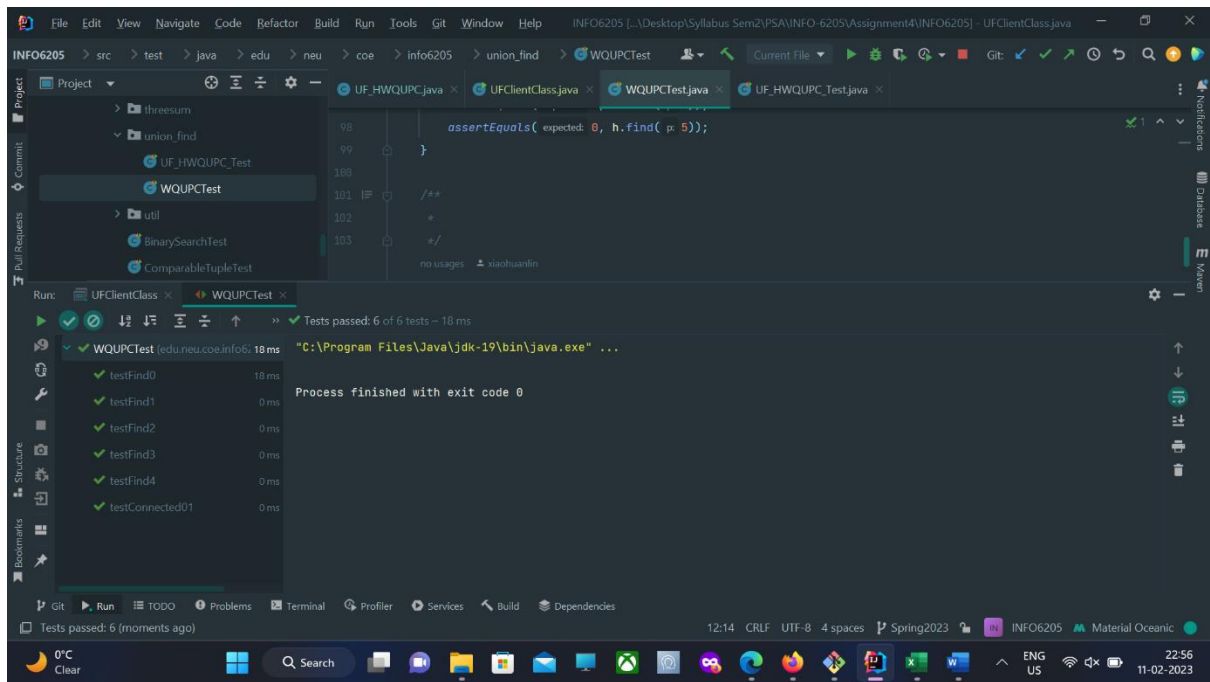
100	183
250	683
500	1626
1000	3179
2000	8090
4000	15614
8000	38949



From the graph above we can conclude that:

The relationship between the number of objects and the number of pairs could be defined with the help of the above graph as a log function of n relation between the two. The value of n is logarithmically related to that of m . The value of n is in direct relation with that of m as $\log(n)$ with some certain constants involved as well.

Unit Test Cases:



```
src > main > java > edu > neu > coe > Info6205 > union_find > UFClientClass > main
Run: UFClientClass
Number of objects
100
objects = 100 with pairs of 183 and connections 99
Number of objects
250
objects = 250 with pairs of 683 and connections 249
Number of objects
500
objects = 500 with pairs of 1626 and connections 499
Number of objects
1000
objects = 1000 with pairs of 3179 and connections 999
Number of objects
2000
objects = 2000 with pairs of 8090 and connections 1999
Number of objects
4000
objects = 4000 with pairs of 15614 and connections 3999
Number of objects
8000
objects = 8000 with pairs of 38949 and connections 7999
Build completed successfully in 2 sec, 402 ms (4 minutes ago)
```