Program Structures and Algorithms

Spring 2023 (SEC – 01)

NAME: Aravind Dasarathy

NUID: 002130918

Task:

- To implement Height Weighted Quick Union with Path Compression (HWQUPC) and to clear all its corresponding unit test cases.
- To create a client that consumes the above implementation and to calculate the number of pairs generated for various sizes of “sites”.
- To deduce a relationship between number of sites vs the number of pairs by creating observations using the above client.

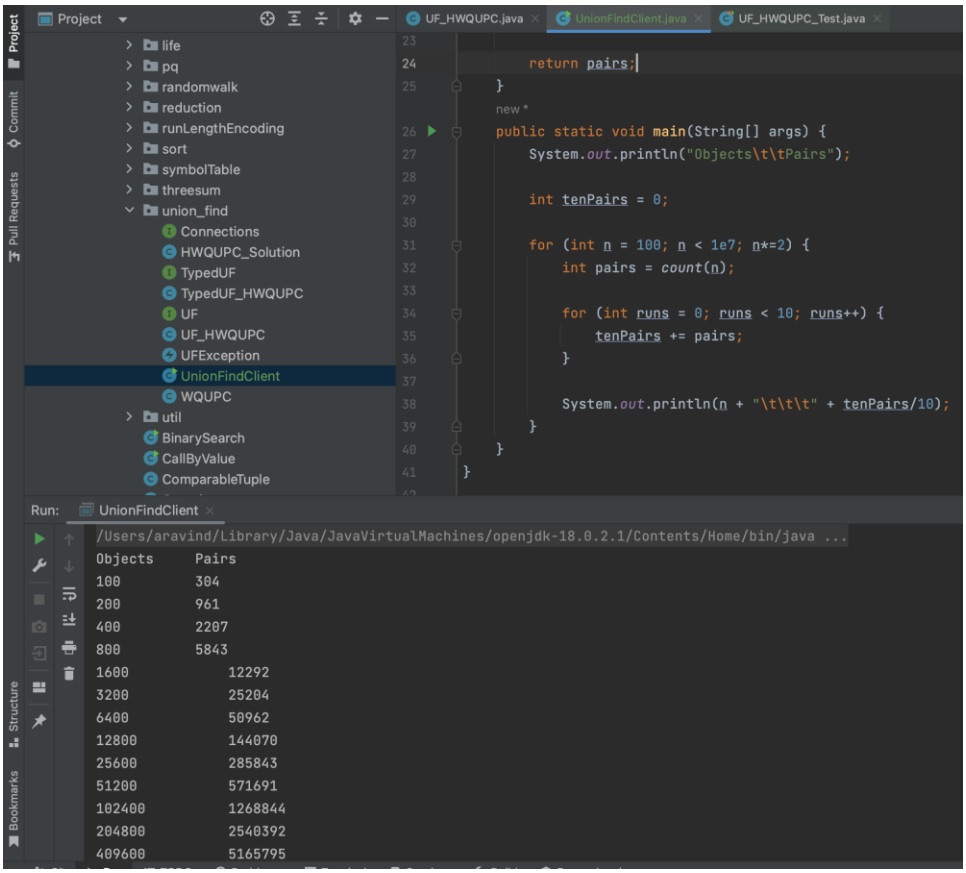
Unit Tests:

All test cases are passed with respect to the HWQUPC project.

Observation:

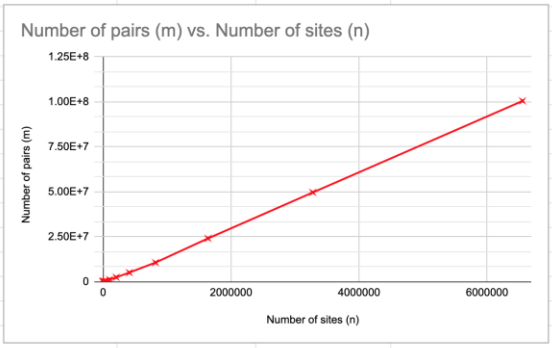
I have implemented an HWQUPC client that consumes the HWQUPC implementation class and have generated several values of the number of pairs ‘m’ for various input sizes of the number of sites ‘n’.

The code to generate the pairs is as follows:



The observation is captured as follows:

Number of sites (n)	Number of pairs (m)
100	304
200	961
400	2207
800	5843
1600	12292
3200	25204
6400	50962
12800	144070
25600	285843
51200	571691
102400	1268844
204800	2540392
409600	5165795
819200	10726210
1638400	24133621
3276800	49687730
6553600	100512178



Logical Conclusion:

From the graph, it can be understood that the ‘n’ is directly proportional to ‘m’.The number of connections,

$$c = n - 1$$

The relationship between ‘n’ and ‘m’ can be given by,

From my code, I’m finding out isConnected(p, q) where ‘p’ and ‘q’ are random numbers from 0 to n-1.

$$m \propto n \cdot \log(n)$$

$$m = k \cdot n \cdot \log(n)$$

In the above equation, the ‘k’ depends on the randomness with which the values ‘p’ and ‘q’ are generated. If a random function ‘rand()’ is used to generate ‘p’ and ‘q’, then, ‘m’ can even approach infinity if the random function is poor. That is, if the rand() generates constant ‘p’ and ‘q’ values all the time, then the number of pairs will approach infinity and the connected components count will never equal to one. Hence, ‘k’ depends on the randomness of the rand().