Parvataneni Mohith lalita Kumar

(NUID : 001565574)

INFO 6205 - Fall 2021

Assignment No : 3

Task :

- To implement height-weighted Quick Union with Path Compression and check that the unit tests for this class all work.
- Using the implementation of UF_HWQUPC, develop a UF client that takes an integer value n from the command line to determine the number of connections.
- To determine the relationship between the number of objects (*n*) and the number of pairs (*m*) generated.

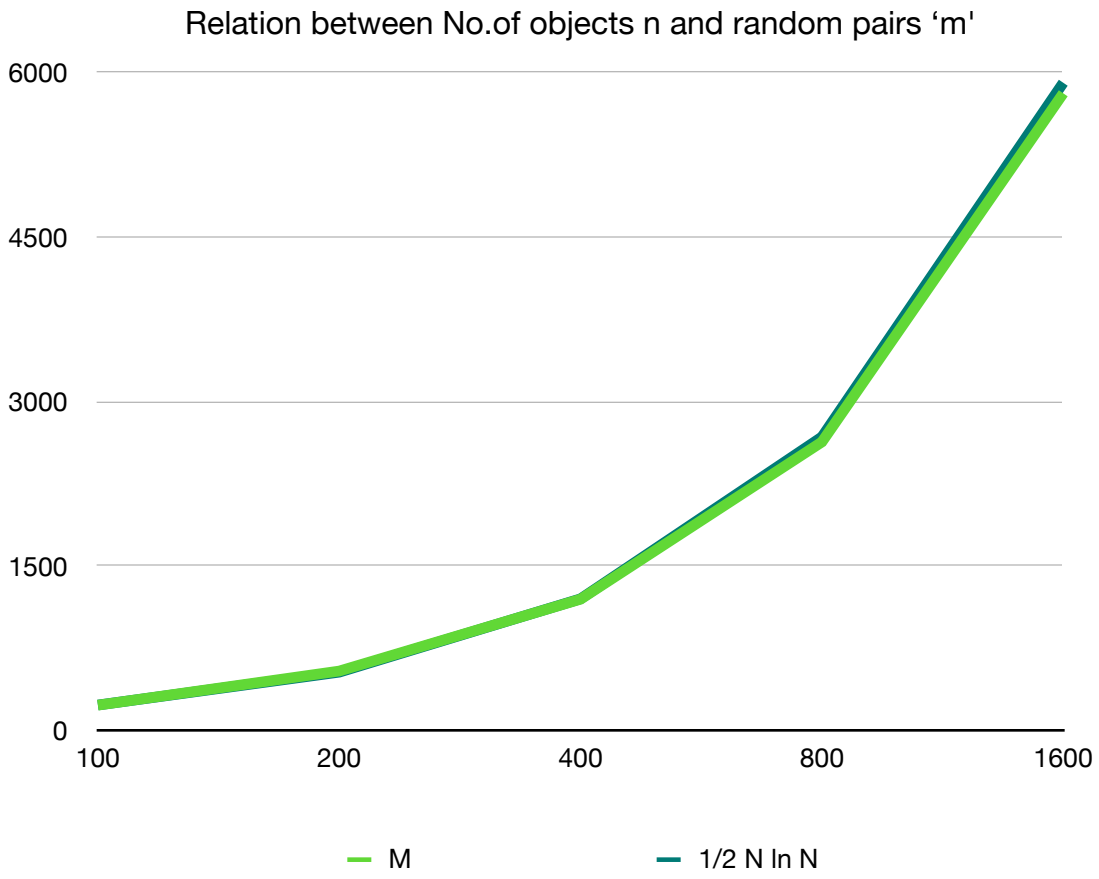
Relationship Conclusion : After performing a series of experiments by passing different arguments of N (100,200,400,800,1600), it can be concluded that the relation between number of random pairs generated (edges) 'M' and the number of objects 'N' is

$$M \approx 1/2\ N \ln N$$

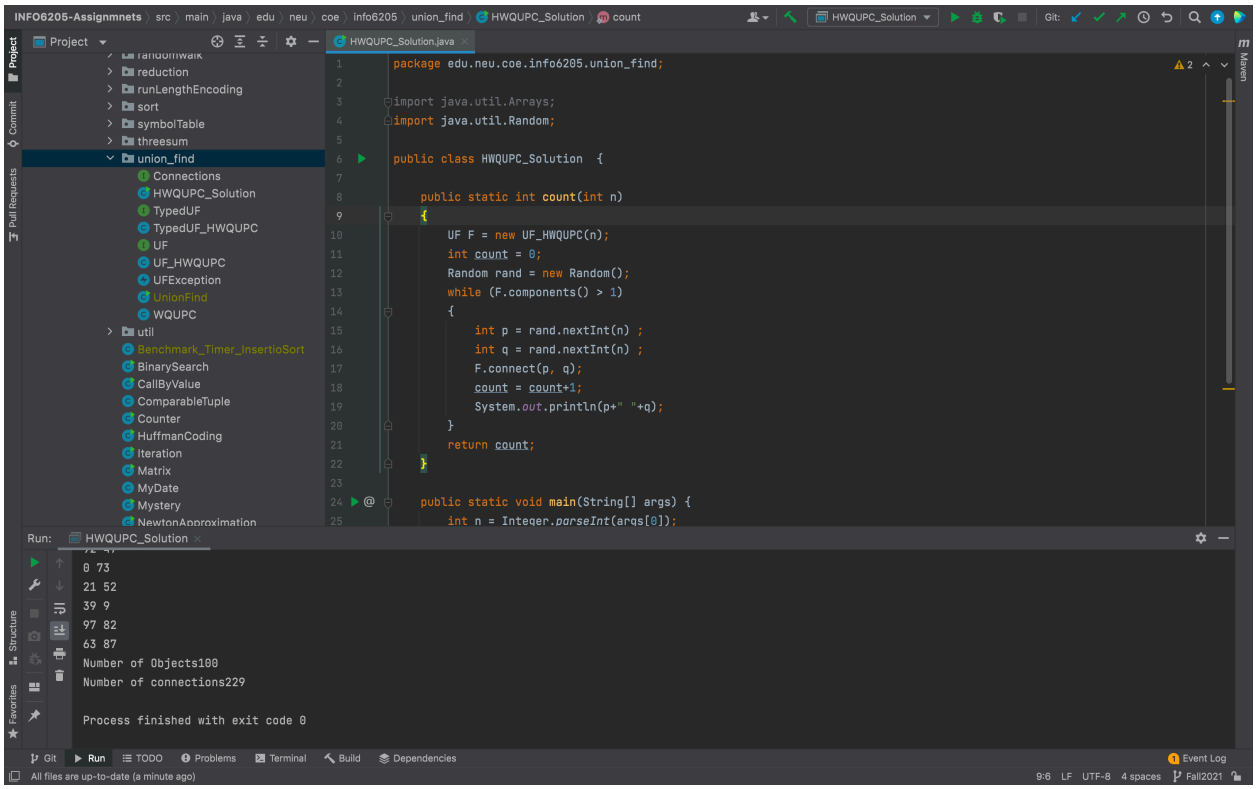
Findings:

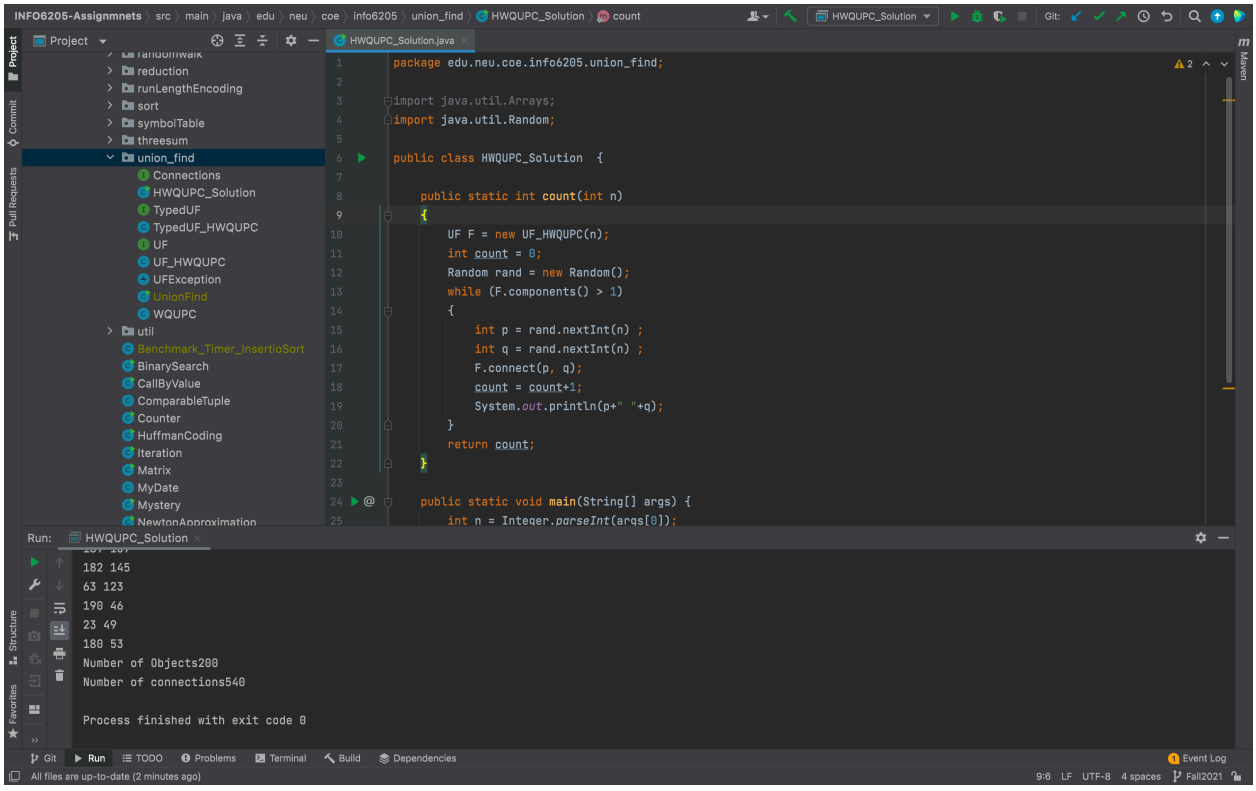
For N = 100, M = 229, 1/2N ln N = 230.25
For N = 200, M = 540, 1/2nN ln N = 529.83
For N = 400, M = 1195, 1/2N ln N = 1198.29
For N = 800, M = 2628, 1/2N ln N = 2673.84
For N = 1600, M = 5806, 1/2N ln N = 5902.20

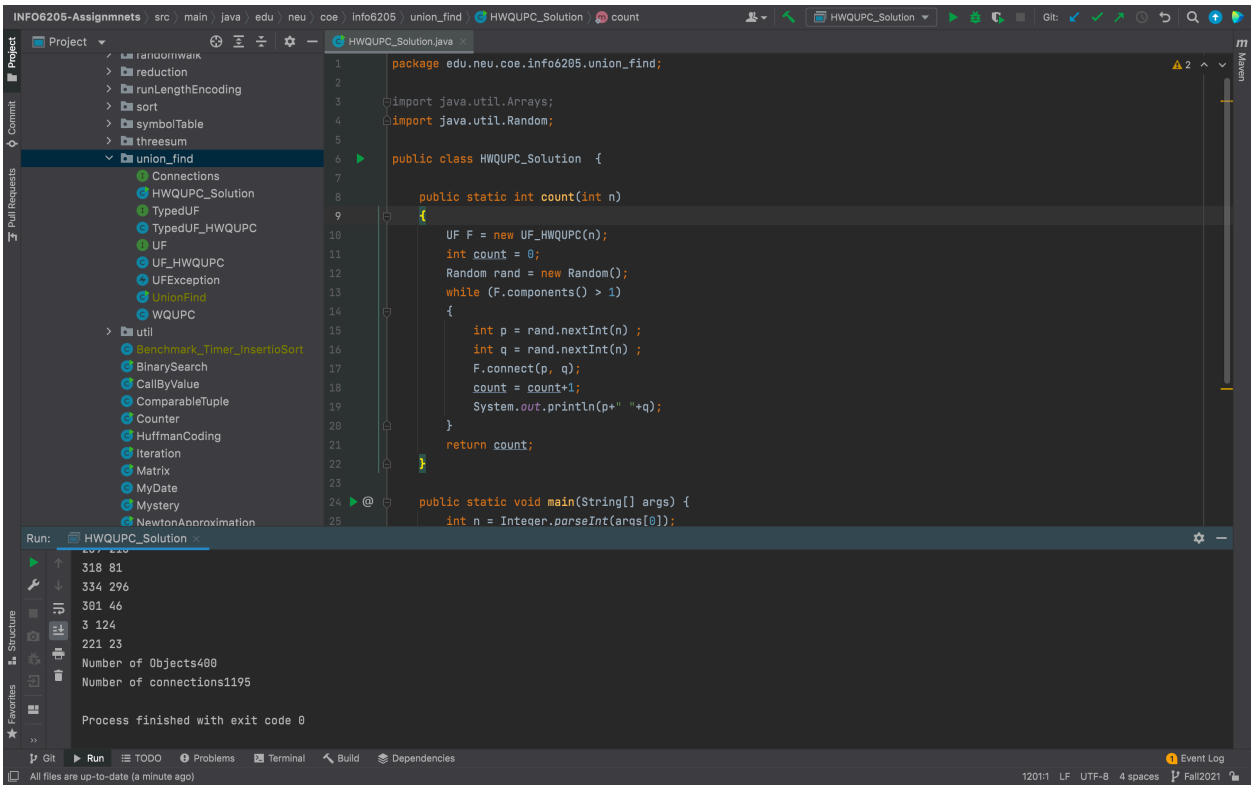
N	M(Actual)	M = 1/2 N ln N (Expected)
10	229	230,25
200	540	529,83
400	1195	1198,29
800	2628	2673,84
1600	5806	5902,20

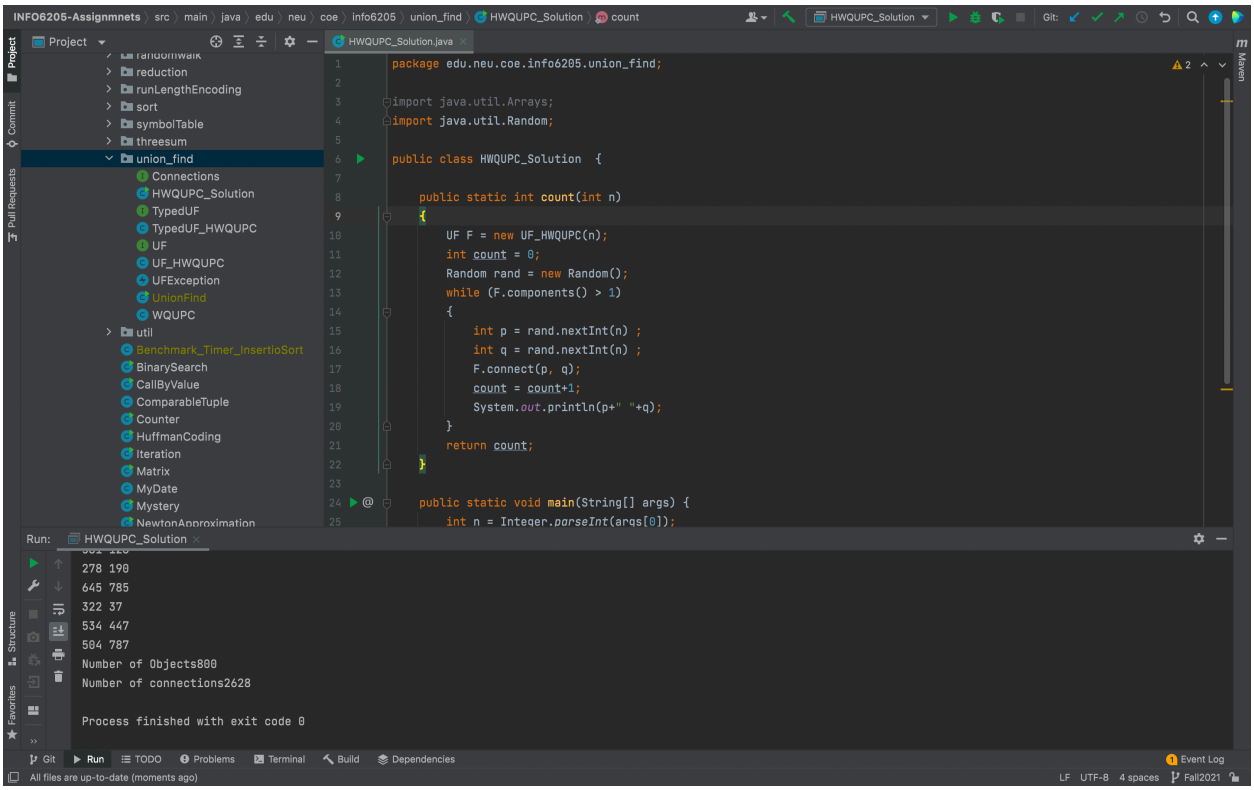


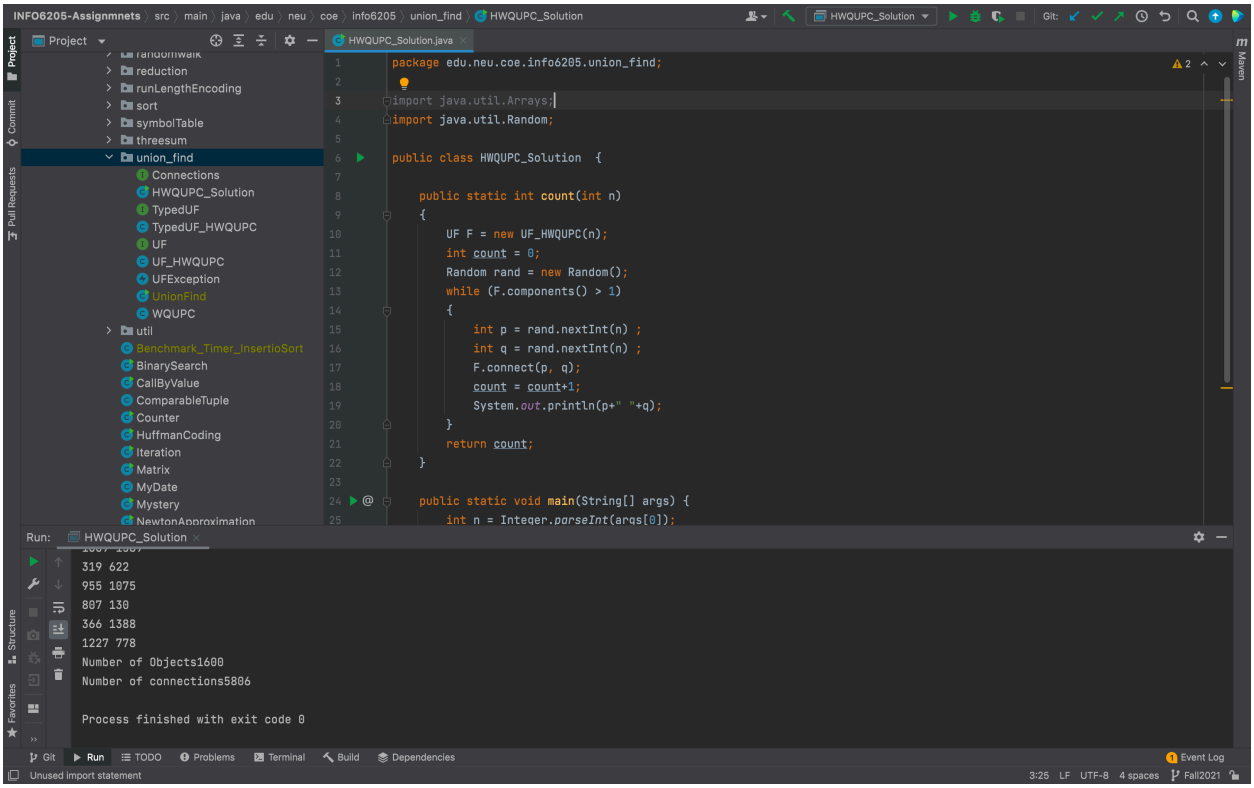
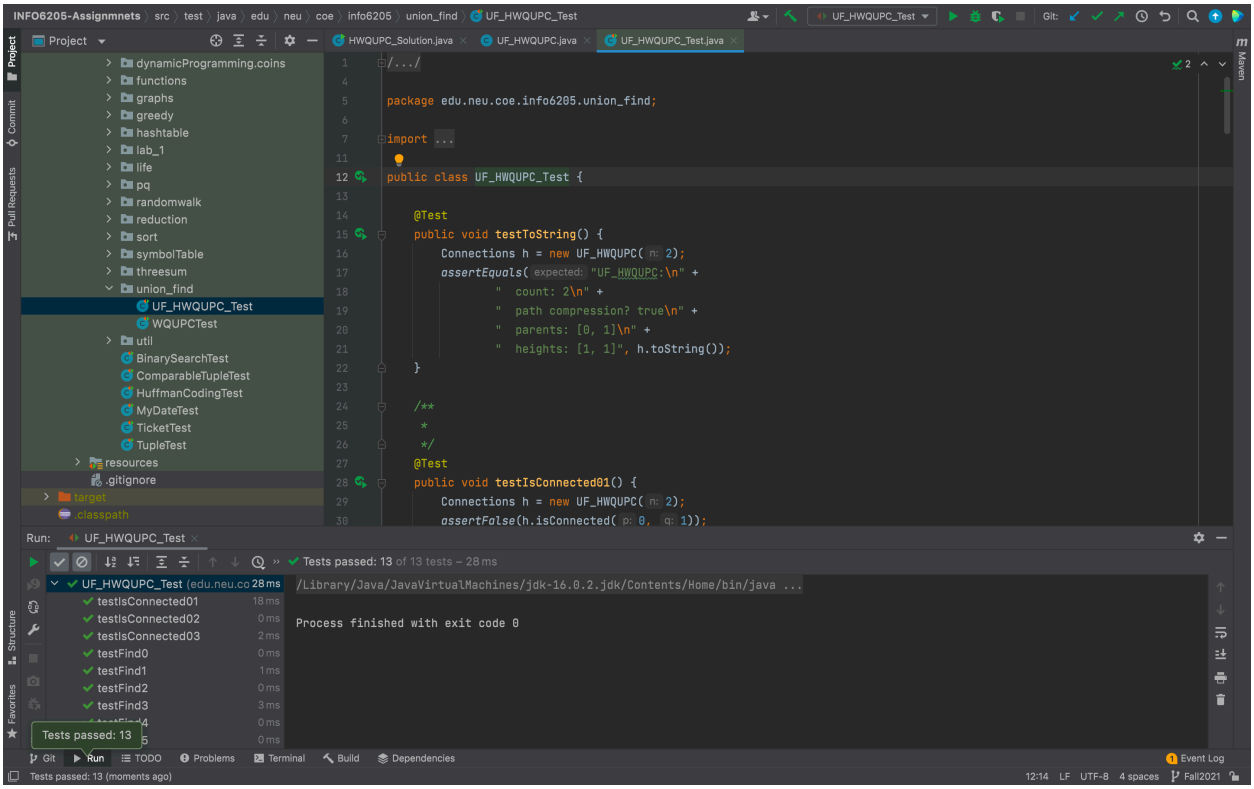
Output Screenshot:






INFO6205-Assignmnets
HWQUPC_Solution.java
package edu.neu.coe.info6205.union_find;
import java.util.Arrays;
import java.util.Random;
public class HWQUPC_Solution {
public static int count(int n)
{
UF F = new UF_HWQUPC(n);
int count = 0;
Random rand = new Random();
while (F.components() > 1)
{
int p = rand.nextInt(n) ;
int q = rand.nextInt(n) ;
F.connect(p, q);
count = count+1;
System.out.println(p+" "+q);
}
return count;
}
public static void main(String[] args) {
int n = Integer.parseInt(args[0]);
Run: HWQUPC_Solution
318 81
334 296
301 46
3 124
221 23
Number of Objects400
Number of connections1195
Process finished with exit code 0


INFO6205-Assignmnets
HWQUPC_Solution.java
package edu.neu.coe.info6205.union_find;
import java.util.Arrays;
import java.util.Random;
public class HWQUPC_Solution {
public static int count(int n)
{
UF F = new UF_HWQUPC(n);
int count = 0;
Random rand = new Random();
while (F.components() > 1)
{
int p = rand.nextInt(n) ;
int q = rand.nextInt(n) ;
F.connect(p, q);
count = count+1;
System.out.println(p+" "+q);
}
return count;
}
public static void main(String[] args) {
int n = Integer.parseInt(args[0]);
Run: HWQUPC_Solution
278 190
645 785
322 37
534 447
504 787
Number of Objects800
Number of connections2628
Process finished with exit code 0



Unit Test Screenshot: