Program Structure and Algorithms

Spring 2022

Assignment 3

Himanshu Walia

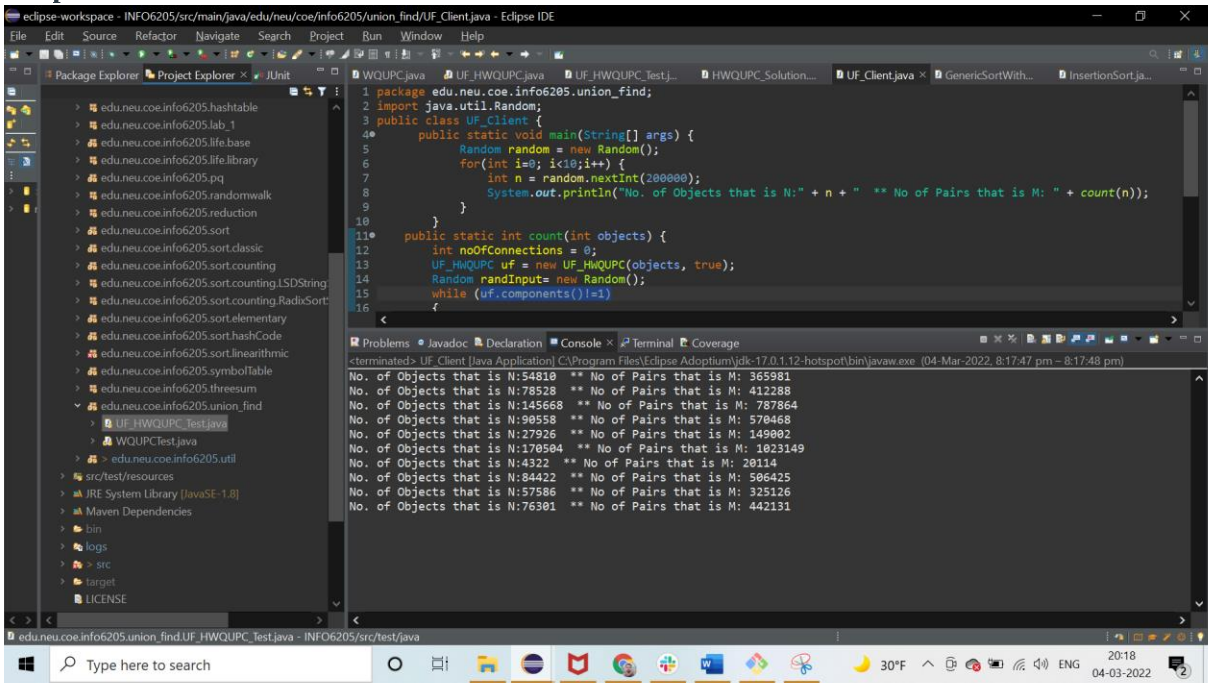
NUID: 002960393

Task: Implement height-weighted Quick Union with Path Compression; changes made in UF_HWQUPC class.

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).

Determine the relationship between M and N.

Output Screenshot:



```java
package edu.neu.coe.info6205.union_find;
import java.util.Random;
public class UF_Client {
	public static void main(String[] args) {
		Random random = new Random();
		for(int i=0; i<10;i++) {
			int n = random.nextInt(200000);
			System.out.println("No. of Objects that is N:" + n + "  ** No of Pairs that is M: " + count(n));
		}
	}
	public static int count(int objects) {
		int noOfConnections = 0;
		UF_HWQUPC uf = new UF_HWQUPC(objects, true);
		Random randInput= new Random();
		while (uf.components()!=1)
		{
```

```
<terminated> UF_Client [Java Application] C:\Program Files\Eclipse Adoptium\jdk-17.0.1.12-hotspot\bin\javaw.exe  (04-Mar-2022, 8:17:47 pm – 8:17:48 pm)
No. of Objects that is N:54810  ** No of Pairs that is M: 365981
No. of Objects that is N:78528  ** No of Pairs that is M: 412288
No. of Objects that is N:145668  ** No of Pairs that is M: 787864
No. of Objects that is N:90558  ** No of Pairs that is M: 570468
No. of Objects that is N:27926  ** No of Pairs that is M: 149002
No. of Objects that is N:170504  ** No of Pairs that is M: 1023149
No. of Objects that is N:4322  ** No of Pairs that is M: 20114
No. of Objects that is N:84422  ** No of Pairs that is M: 506425
No. of Objects that is N:57586  ** No of Pairs that is M: 325126
No. of Objects that is N:76301  ** No of Pairs that is M: 442131
```

Output in the console Window(Eclipse):

```
No. of Objects that is N:54810  ** No of Pairs that is M: 365981
No. of Objects that is N:78528  ** No of Pairs that is M: 412288
No. of Objects that is N:145668  ** No of Pairs that is M: 787864
No. of Objects that is N:90558  ** No of Pairs that is M: 570468
No. of Objects that is N:27926  ** No of Pairs that is M: 149002
No. of Objects that is N:170504  ** No of Pairs that is M: 1023149
No. of Objects that is N:4322  ** No of Pairs that is M: 20114
No. of Objects that is N:84422  ** No of Pairs that is M: 506425
No. of Objects that is N:57586  ** No of Pairs that is M: 325126
No. of Objects that is N:76301  ** No of Pairs that is M: 442131
```

Relationship/Conclusion: It is quite evident from the above outputs that No. of pairs generated that are M and Number of objects that are N are proportional.

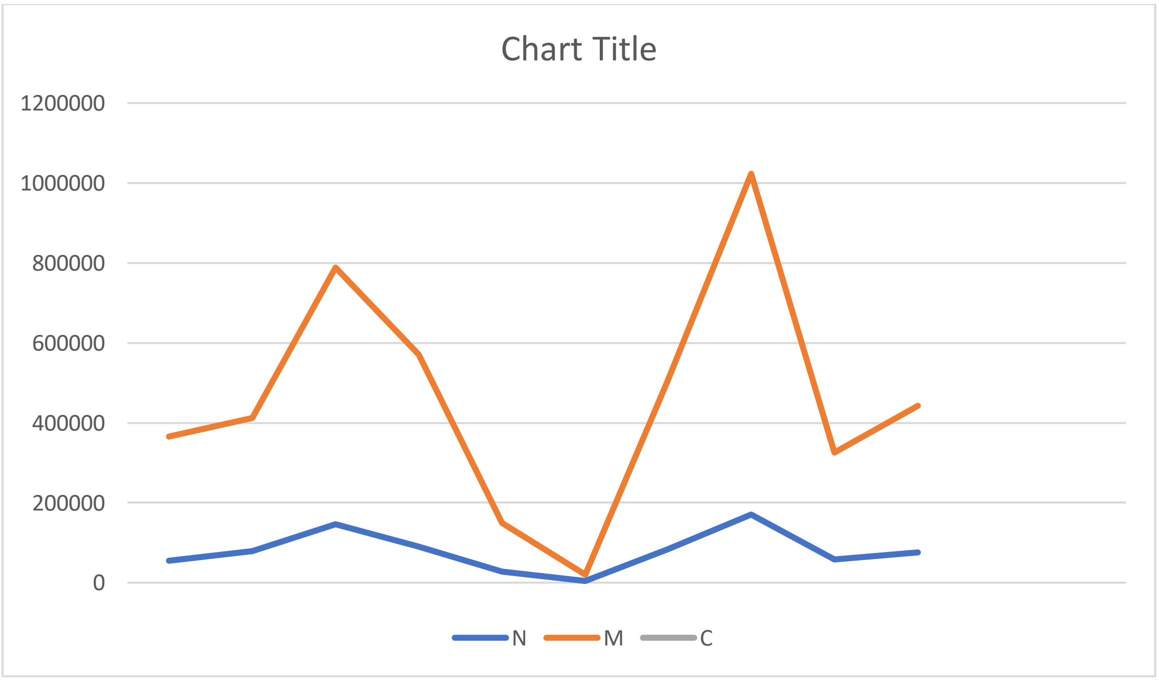
Taking the average of all the 10 cases it comes out to be around M~4*N.

Evidence: Below are the table and a graph to prove the relationship depicted above.

It is clearly shown from the proportional increase in the value of M with respect to the value of N.

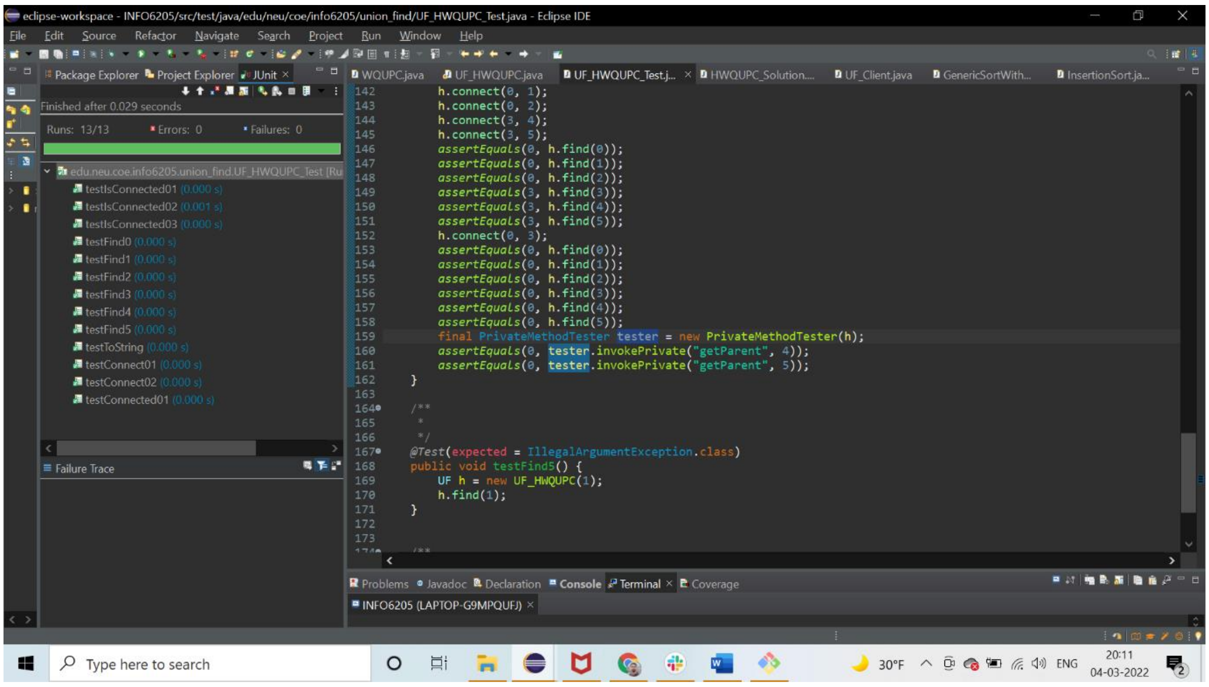
Number of Object: N	Number of Pairs: M
54810	365981
78528	412288
145668	787864
90558	570468
27926	149002
4322	20114
84422	506425
170504	1023149
57586	325126
76301	442131

In the below graph the value on the left side (Y -axis) represents the number of Objects(N).



Screenshot of the Unit Test case Passed:

Please find the screenshot showing all the test case passing for the UF_HWQUPC_Test.java



eclipse-workspace - INFO6205/src/test/java/edu/neu/coe/info6205/union_find/UF_HWQUPC_Test.java - Eclipse IDE

```
Finished after 0.029 seconds

Runs: 13/13     Errors: 0     Failures: 0

edu.neu.coe.info6205.union_find.UF_HWQUPC_Test [Ru
    testIsConnected01 (0.000 s)
    testIsConnected02 (0.001 s)
    testIsConnected03 (0.000 s)
    testFind0 (0.000 s)
    testFind1 (0.000 s)
    testFind2 (0.000 s)
    testFind3 (0.000 s)
    testFind4 (0.000 s)
    testFind5 (0.000 s)
    testToString (0.000 s)
    testConnect01 (0.000 s)
    testConnect02 (0.000 s)
    testConnected01 (0.000 s)
```

```java
        h.connect(0, 1);
        h.connect(0, 2);
        h.connect(3, 4);
        h.connect(3, 5);
        assertEquals(0, h.find(0));
        assertEquals(0, h.find(1));
        assertEquals(0, h.find(2));
        assertEquals(3, h.find(3));
        assertEquals(3, h.find(4));
        assertEquals(3, h.find(5));
        h.connect(0, 3);
        assertEquals(0, h.find(0));
        assertEquals(0, h.find(1));
        assertEquals(0, h.find(2));
        assertEquals(0, h.find(3));
        assertEquals(0, h.find(4));
        assertEquals(0, h.find(5));
        final PrivateMethodTester tester = new PrivateMethodTester(h);
        assertEquals(0, tester.invokePrivate("getParent", 4));
        assertEquals(0, tester.invokePrivate("getParent", 5));
    }

    /**
     *
     */
    @Test(expected = IllegalArgumentException.class)
    public void testFind5() {
        UF h = new UF_HWQUPC(1);
        h.find(1);
    }


    /**
```