Program Structures & Algorithms

Spring 2022

Assignment 3 (WQUPC)

Name: Aditya Kinare

(NUID): 001095881

- **Task**
- **Output screenshot**
- **Relationship Conclusion**
- **Evidence / Graph**
- **Unit tests result**

Task:

1. Implement height-weighted Quick Union with Path Compression.
2. Use WQUPC class to get multiple values of generated pairs for N components
3. Derive a relation between number of components(N), and number of Pairs(M) generated to create a cyclic graph.

Output screenshot:

Code Implementation for UF_HWQUPC

```
81    public int find(int p) {
82        validate(p);
83        int root = p;
84        while(root != parent[root]) {
85            if (this.pathCompression) {
86                doPathCompression(root);
87            }
88            root = parent[root];
89        }
90        // END
91        return root;
92    }
93
```

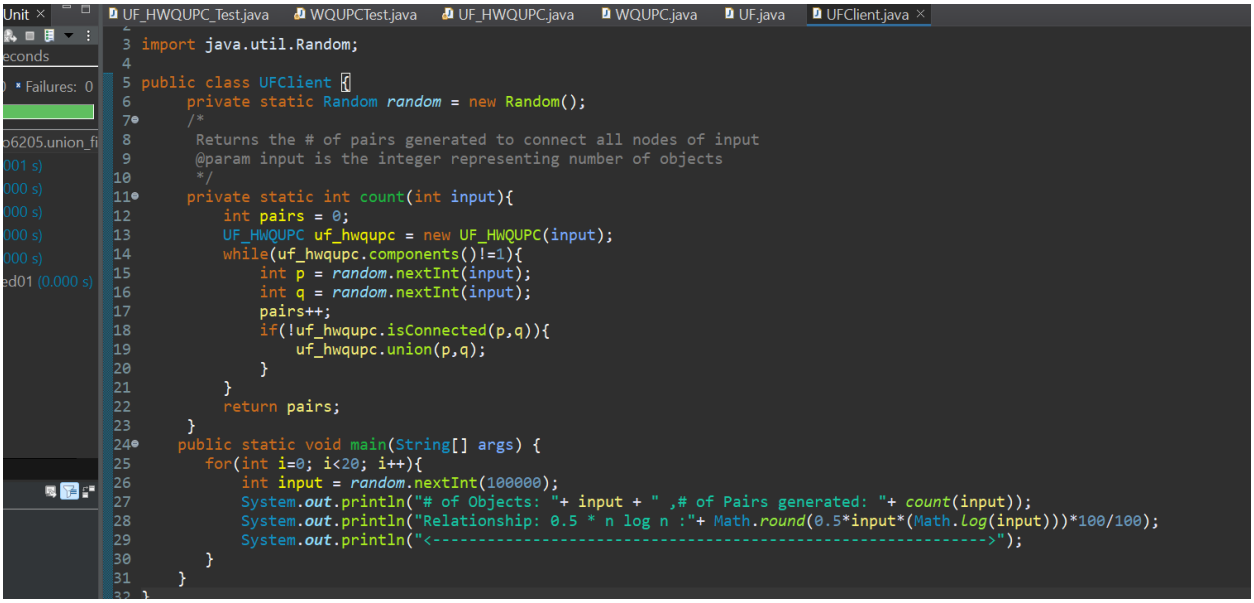
```java
private void doPathCompression(int i) {
    // FIXME update parent to value of grandparent
    parent[i] = parent[parent[i]];
    // END
}
```

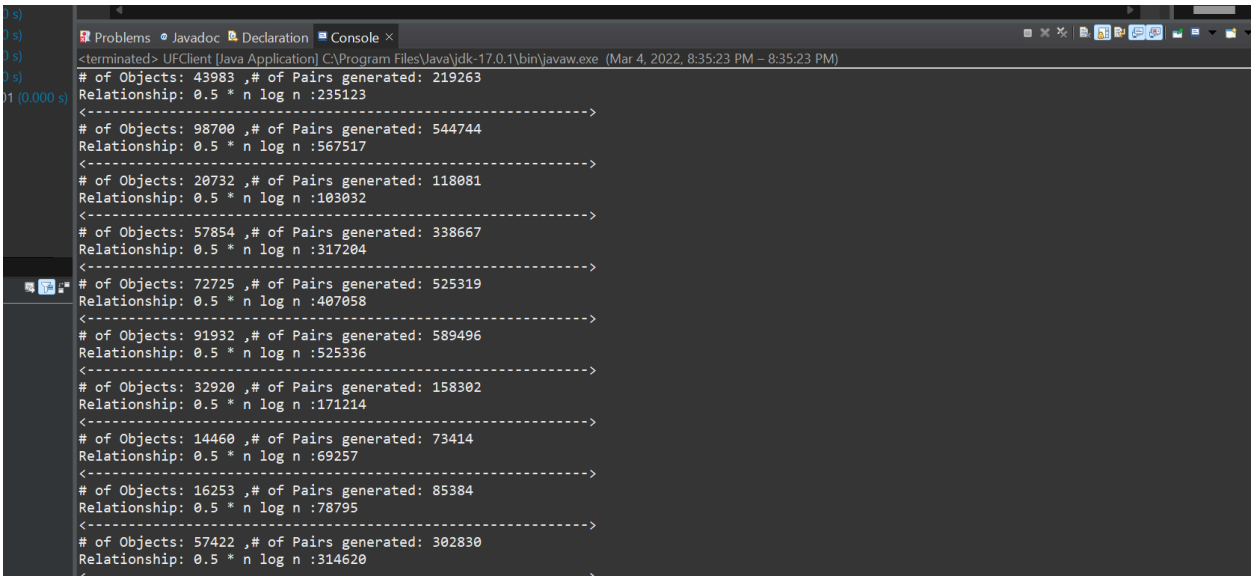
```java
private void mergeComponents(int i, int j) {
    // FIXME make shorter root point to taller one
    int a = parent[i];
    int b = parent[j];

    if(height[a] < height[b]) {
        parent[a] = b;
        height[b] += height[a];

    }else{
        parent[b] = parent[a];
        height[a] += height [b];
    }
    // END
}
```

Implemented the UFClient.java class to get evidence for deriving a relation between N and M



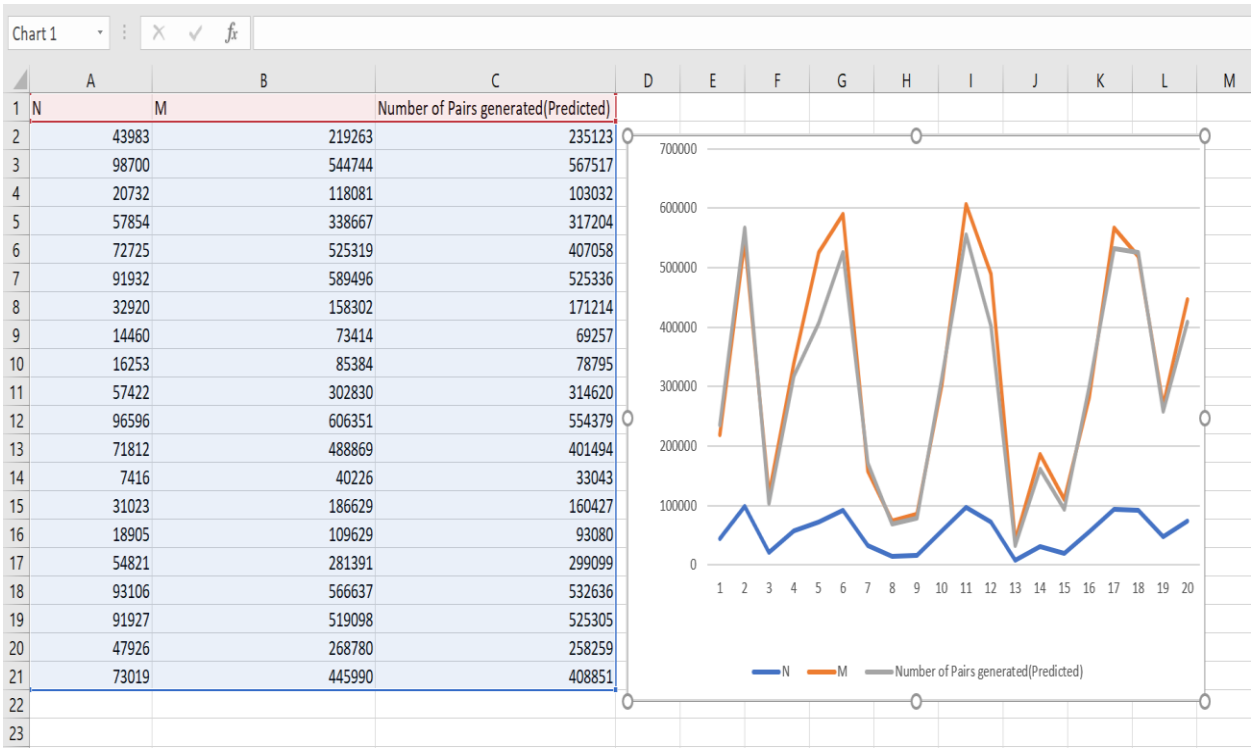


Relationship Conclusion: The relationship between number of Objects (n) and number of pairs (m) generated to reach to single component i.e all objects should be connected is mentioned below:

Let n be number of objects and m be number of pairs generated to connect those objects

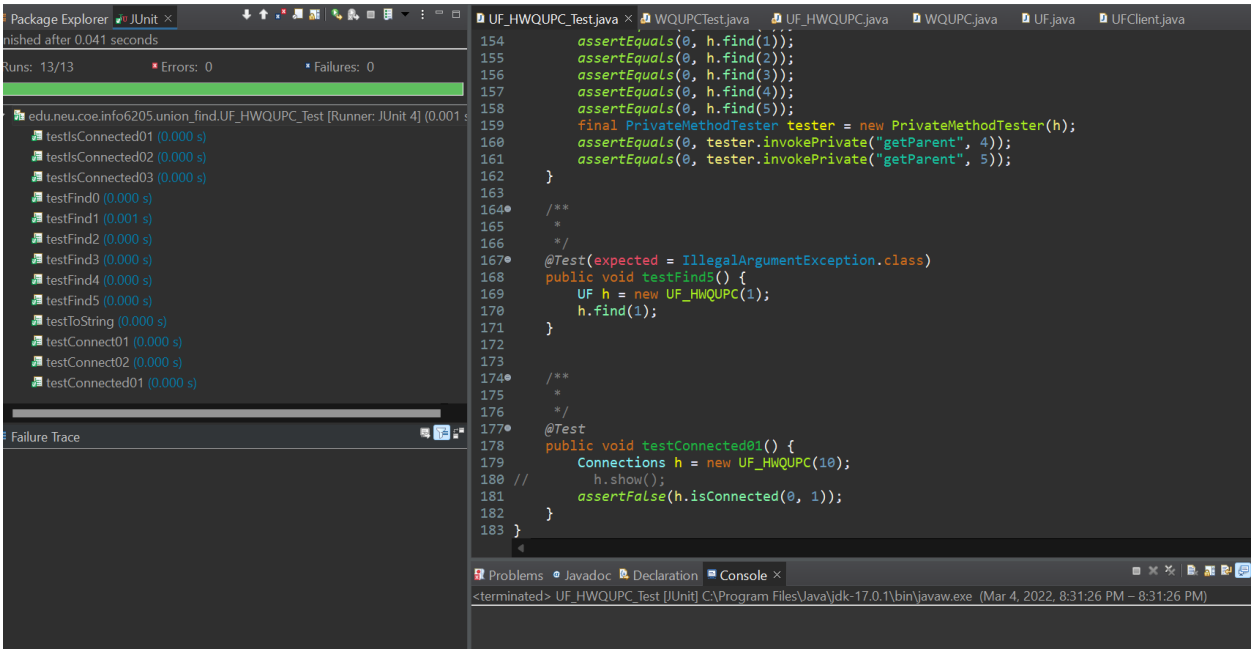
m = 0.5 * n log n

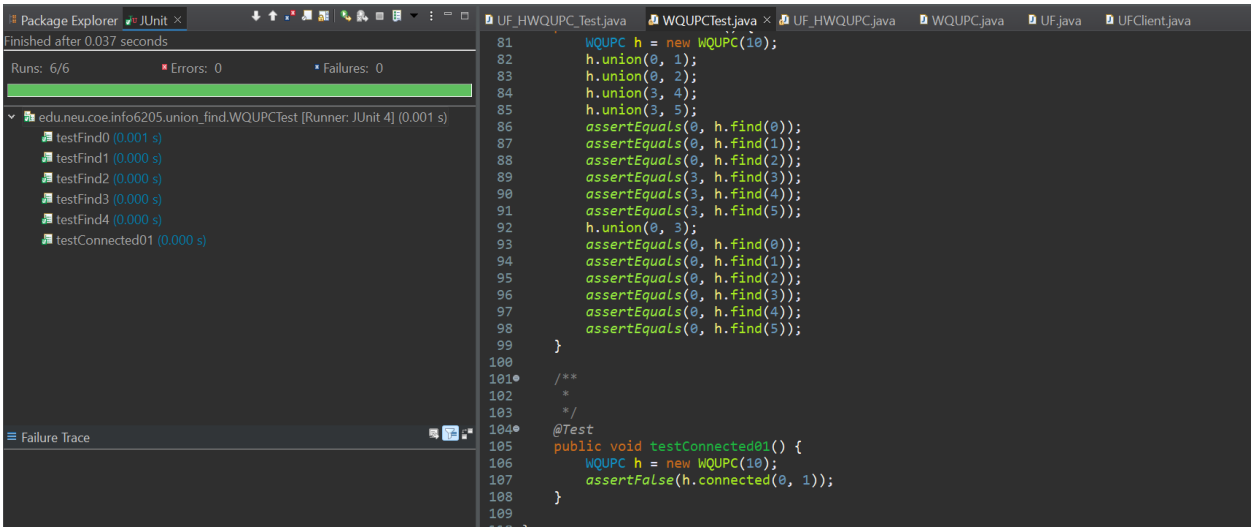
Evidence/Graph:



N	M	Number of Pairs generated(Predicted)
43983	219263	235123
98700	544744	567517
20732	118081	103032
57854	338667	317204
72725	525319	407058
91932	589496	525336
32920	158302	171214
14460	73414	69257
16253	85384	78795
57422	302830	314620
96596	606351	554379
71812	488869	401494
7416	40226	33043
31023	186629	160427
18905	109629	93080
54821	281391	299099
93106	566637	532636
91927	519098	525305
47926	268780	258259
73019	445990	408851

Unit Test Results:




Package Explorer
JUnit
Finished after 0.037 seconds
Runs: 6/6
Errors: 0
Failures: 0
edu.neu.coe.info6205.union_find.WQUPCTest [Runner: JUnit 4] (0.001 s)
testFind0 (0.001 s)
testFind1 (0.000 s)
testFind2 (0.000 s)
testFind3 (0.000 s)
testFind4 (0.000 s)
testConnected01 (0.000 s)
Failure Trace
UF_HWQUPC_Test.java
WQUPCTest.java
UF_HWQUPC.java
WQUPC.java
UF.java
UFClient.java
WQUPC h = new WQUPC(10);
h.union(0, 1);
h.union(0, 2);
h.union(3, 4);
h.union(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.union(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));
}
/**
*
*/
@Test
public void testConnected01() {
WQUPC h = new WQUPC(10);
assertFalse(h.connected(0, 1));
}