

**Sutiksh Verma**

**002122052**

**INFO 6205**

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No. 3**

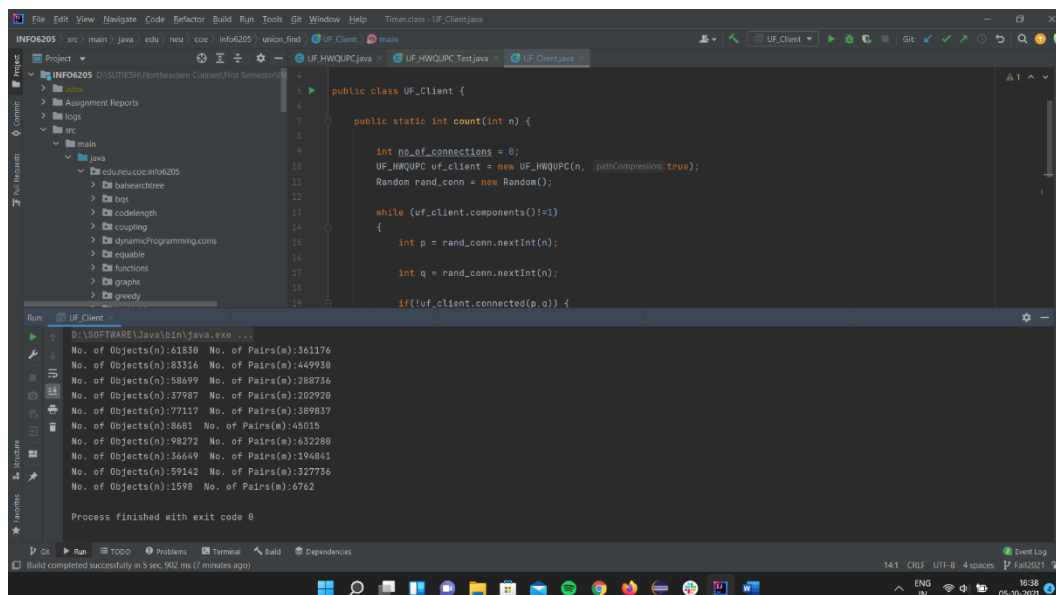
**Task:** To implement height-weighted Quick union with Path compression.

For this task UF\_HWQUPC java class was used and following methods were implemented

- find () – method to update the root of input object if pathCompression is performed
- mergeComponents() – method to merge 2 subtrees such that smaller root points to larger root
- doPathCompression() – method that implements the single-pass process of pathCompression method.

Also, UF\_Client java class was created to perform and test the implementation of UF\_HWQUPC class

**Output:**



```
public class UF_Client {  
    public static int count(int n) {  
        int no_of_connections = 0;  
        UF_HWQUPC uf_client = new UF_HWQUPC(n, pathCompression: true);  
        Random rand_conn = new Random();  
        while (uf_client.components() != 1) {  
            int p = rand_conn.nextInt(n);  
            int q = rand_conn.nextInt(n);  
            if (!uf_client.connected(p, q)) {  
                no_of_connections++;  
            }  
        }  
        return no_of_connections;  
    }  
}
```

Run

0:\SOFTWARE\Java\bin\java.exe ...  
No. of Objects(n):61838 No. of Pairs(m):361174  
No. of Objects(n):83316 No. of Pairs(m):449938  
No. of Objects(n):58699 No. of Pairs(m):288736  
No. of Objects(n):37987 No. of Pairs(m):282928  
No. of Objects(n):77117 No. of Pairs(m):389837  
No. of Objects(n):8681 No. of Pairs(m):45815  
No. of Objects(n):98272 No. of Pairs(m):632288  
No. of Objects(n):36649 No. of Pairs(m):194841  
No. of Objects(n):59142 No. of Pairs(m):327736  
No. of Objects(n):1598 No. of Pairs(m):6762  
Process finished with exit code 0

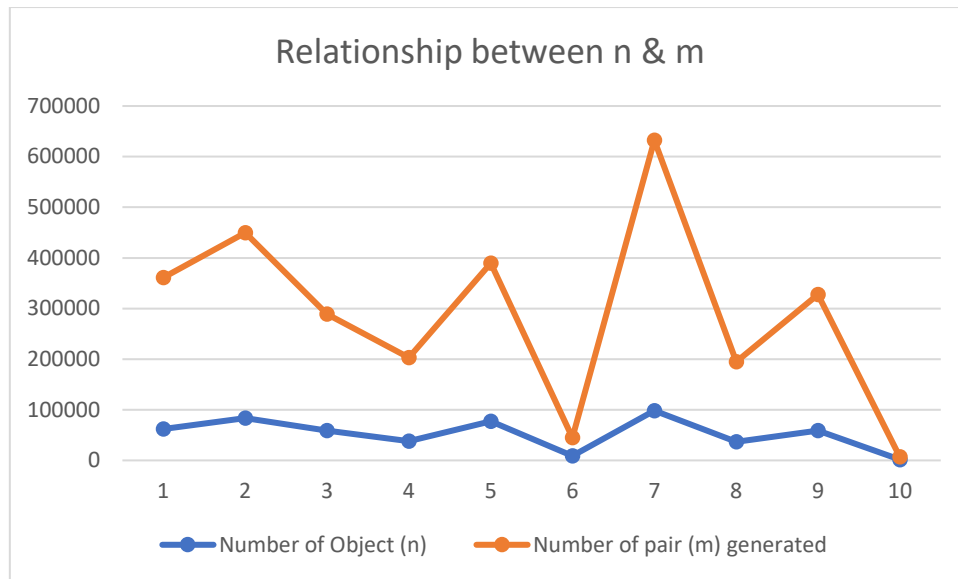
### Console Output:

No. of Objects(n):61830 No. of Pairs(m):361176  
No. of Objects(n):83316 No. of Pairs(m):449930  
No. of Objects(n):58699 No. of Pairs(m):288736  
No. of Objects(n):37987 No. of Pairs(m):202920  
No. of Objects(n):77117 No. of Pairs(m):389837  
No. of Objects(n):8681 No. of Pairs(m):45015  
No. of Objects(n):98272 No. of Pairs(m):632280  
No. of Objects(n):36649 No. of Pairs(m):194841  
No. of Objects(n):59142 No. of Pairs(m):327736  
No. of Objects(n):1598 No. of Pairs(m):6762

**Relationship:** It can be concluded from the results mentioned above that the number of pairs(m) generated are proportional to the number of objects provided as input. i.e  $m \sim 5 * n$

**Evidence:** I have attached a table and a chart to show the relationship between the number of object (n) and number of pair (m) generated with different set of values for both n and m. As a result, we can see the proportionate result between n and m.

Number of Object (n)	Number of pair (m) generated
61830	361176
83316	449930
58699	288736
37987	202920
77117	389837
8681	45015
98272	632280
36649	194841
59142	327736
1598	6762



The left side of the chart is the value of object (n)

**Screenshots of Passed Unit tests:** I have attached the screenshot of successfully passed unit test for the class `UF_HWQUPC` test class.

### UF\_HWQUPC\_Test.java

```

public class UF_HWQUPC_Test {

    @Test
    public void testToString() {
        Connections h = new UF_HWQUPC(10, 2);
        assertEquals("expected: \"UF_HWQUPC:\\n\" +\n" +
            "    count: 2\\n\" +\n" +
            "    path compression? true\\n\" +\n" +
            "    parents: [0, 1]\\n\" +\n" +
            "    heights: [1, 1]\", h.toString());
    }
}

```

Run: UF\_HWQUPC\_Test

Tests passed: 13 of 13 tests - 0 ms

UF\_HWQUPC\_Test (edu.neu.coe.info6205.u 0 ms)

- ✓ testsConnected01 0 ms
- ✓ testsConnected02 0 ms
- ✓ testsConnected03 0 ms
- ✓ testFind0 0 ms
- ✓ testFind1 0 ms
- ✓ testFind2 0 ms
- ✓ testFind3 0 ms
- ✓ testFind4 0 ms
- ✓ testFind5 0 ms
- ✓ testToString 0 ms
- ✓ testConnect01 0 ms
- ✓ testConnect02 0 ms
- ✓ testConnect01 0 ms

Process finished with exit code 0