

# Akshaysingh Bayes (NU ID: 002956209)

## 6205 - Program Structures and Algorithms

### Assignment - 3

#### Problem Statement:

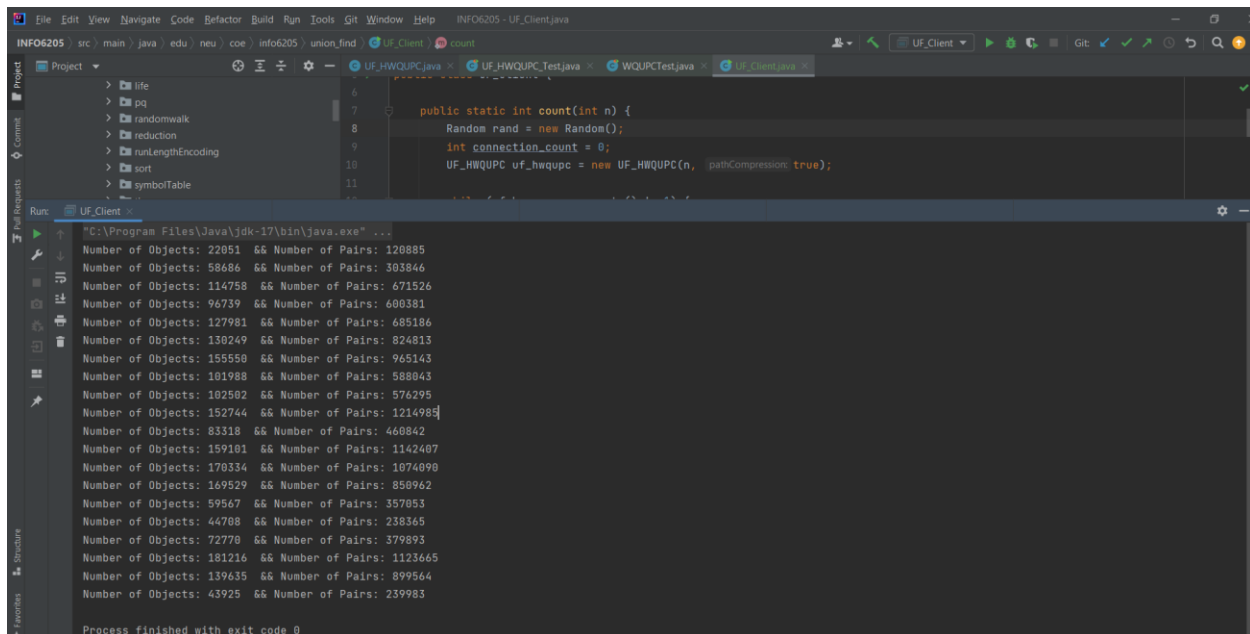
To implement height-weighted Quick Union with path compression. For this task UF\_HWQUPC java class was used and methods that were implemented:

1. find ()
2. mergeComponents ()
3. doPathCompression ()

A new class UF\_Client was implemented.

#### Output:

UF\_Client main () ran successfully. It ran for 20 times, printing following output in console.



```
INFO6205 - UF_Client.java
src \ main \ java \ edu \ neu \ coe \ info6205 \ union_find \ UF_Client \ count
UF_Client
UF_HWQUPC.java
UF_HWQUPC_Test.java
WQUPCTest.java
UF_Client.java

public static int count(int n) {
    Random rand = new Random();
    int connection_count = 0;
    UF_HWQUPC uf_hwqupc = new UF_HWQUPC(n, pathCompression: true);

    for (int i = 0; i < n; i++) {
        int p1 = rand.nextInt(n);
        int p2 = rand.nextInt(n);
        if (p1 != p2) {
            uf_hwqupc.merge(p1, p2);
            connection_count++;
        }
    }
    return connection_count;
}

Run: UF_Client
C:\Program Files\Java\jdk-17\bin\java.exe ...
Number of Objects: 22051 && Number of Pairs: 120885
Number of Objects: 58686 && Number of Pairs: 303846
Number of Objects: 114758 && Number of Pairs: 671526
Number of Objects: 96739 && Number of Pairs: 600381
Number of Objects: 127981 && Number of Pairs: 685186
Number of Objects: 130249 && Number of Pairs: 824813
Number of Objects: 155550 && Number of Pairs: 965143
Number of Objects: 101988 && Number of Pairs: 588043
Number of Objects: 102502 && Number of Pairs: 576295
Number of Objects: 152744 && Number of Pairs: 1214985
Number of Objects: 83318 && Number of Pairs: 460842
Number of Objects: 159101 && Number of Pairs: 1142407
Number of Objects: 170334 && Number of Pairs: 1074090
Number of Objects: 169529 && Number of Pairs: 858962
Number of Objects: 59567 && Number of Pairs: 357053
Number of Objects: 44708 && Number of Pairs: 238365
Number of Objects: 72770 && Number of Pairs: 379893
Number of Objects: 181216 && Number of Pairs: 1123665
Number of Objects: 139635 && Number of Pairs: 899544
Number of Objects: 43925 && Number of Pairs: 239983

Process finished with exit code 0
```

#### Console Output:

Number of Objects: 22051 && Number of Pairs: 120885

Number of Objects: 58686 && Number of Pairs: 303846

Number of Objects: 114758 && Number of Pairs: 671526

Number of Objects: 96739 && Number of Pairs: 600381

Number of Objects: 127981 && Number of Pairs: 685186

Number of Objects: 130249 && Number of Pairs: 824813

Number of Objects: 155550 && Number of Pairs: 965143

Number of Objects: 101988 && Number of Pairs: 588043

Number of Objects: 102502 && Number of Pairs: 576295

Number of Objects: 152744 && Number of Pairs: 1214985

Number of Objects: 83318 && Number of Pairs: 460842

Number of Objects: 159101 && Number of Pairs: 1142407

Number of Objects: 170334 && Number of Pairs: 1074090

Number of Objects: 169529 && Number of Pairs: 850962

Number of Objects: 59567 && Number of Pairs: 357053

Number of Objects: 44708 && Number of Pairs: 238365

Number of Objects: 72770 && Number of Pairs: 379893

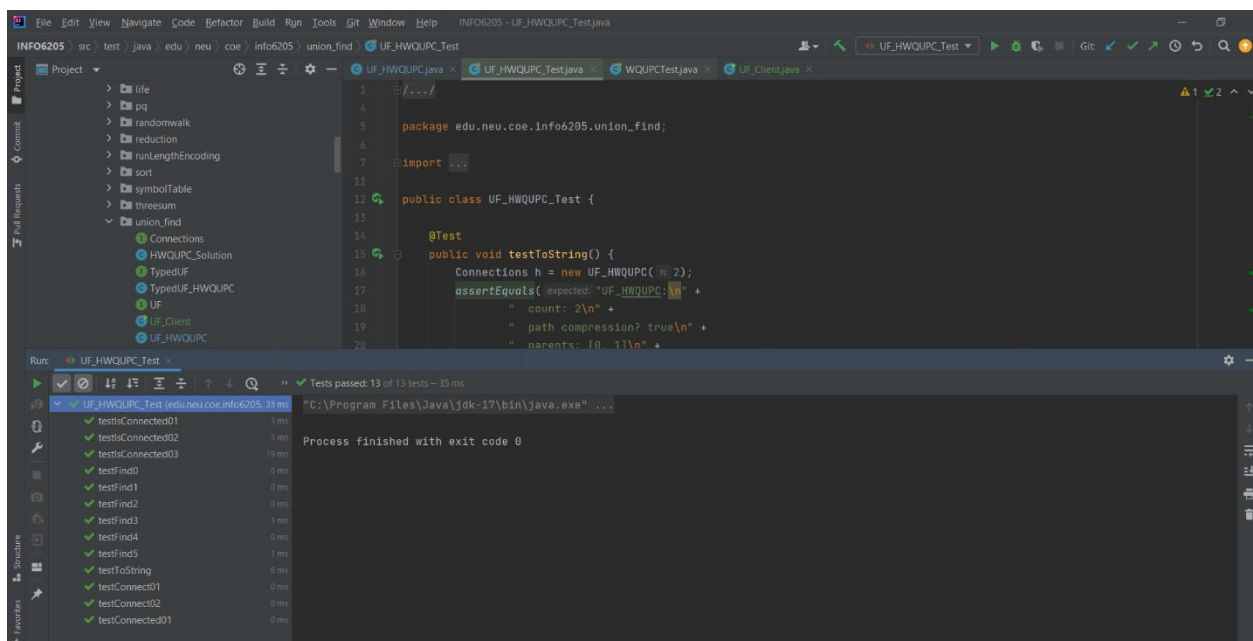
Number of Objects: 181216 && Number of Pairs: 1123665

Number of Objects: 139635 && Number of Pairs: 899564

Number of Objects: 43925 && Number of Pairs: 239983

## Test Case Results:

UF\_HWQUPC.java ran successfully, passing all tests.



```
INFO6205 - UF_HWQUPC_Test.java
src \ test \ java \ edu \ neu \ coe \ info6205 \ union_find \ UF_HWQUPC_Test
UF_HWQUPC_Test
UF_HWQUPC_Test.java
package edu.neu.coe.info6205.union_find;
import ...
public class UF_HWQUPC_Test {
    @Test
    public void testToString() {
        Connections h = new UF_HWQUPC(10, 2);
        assertEquals("expected: \"UF_HWQUPC: 10 2\" +
            \" count: 2\\n\" +
            \" path compression? true\\n\" +
            \" parents: {0..1}\\n\" +",
            h.toString());
    }
}
Run: UF_HWQUPC_Test
Tests passed: 13 of 13 tests - 35 ms
UF_HWQUPC_Test (edu.neu.coe.info6205, 13 ms)
testisConnected01 3 ms
testisConnected02 3 ms
testisConnected03 10 ms
testisInd0 0 ms
testisInd1 0 ms
testisInd2 0 ms
testisInd3 1 ms
testisInd4 0 ms
testisInd5 1 ms
testToString 0 ms
testConnect01 0 ms
testConnect02 0 ms
testConnect03 0 ms
Process finished with exit code 0
```

## Deduction:

After reviewing results of the following experiment, it can be concluded that:

Number of Pairs (m) generated is directly proportional to the Number of Objects (n) that are provided as input i.e.

$$m \sim 6*n$$

## Practical Evidence:

Following table depict the relationship between the number of objects(n) and the number of pairs(m) generated when ran 20 times with randomly selected values of n.

Index	Number of Objects (n)	Number of Pairs (m)
1	22051	120885
2	58686	303846
3	114758	671526
4	96739	600381
5	127981	685186
6	130249	824813
7	155550	965143
8	101988	588043
9	102502	576295
10	152744	1214985
11	83318	460842
12	159101	1142407
13	170334	1074090
14	169529	850962
15	59567	357053
16	44708	238365
17	72770	379893
18	181216	1123665
19	139635	899564
20	43925	239983

## Charts:

The same relationship can be depicted using the following chart plotted below.

## Logarithmic Chart

