Assignment No. 3

Goal

- (Part 1) Implement height-weighted Quick Union with Path Compression

- (Part 2) Develop a UF 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.

- (Part 3) Determine the relationship between the number of objects (o) and the number of pairs (p)

Relationship Conclusion:

The concluded relationship between the number of objects (o) and the number of pairs (p) generated for the reduction of the number of components from o to 1 is:

$$\mathrm{p} = f(\mathrm{o}) = \frac{1}{2} \times \mathrm{o} \times \ln(\mathrm{o})$$

Where,

$$\mathrm{p} = number\ of\ pairs\ generated\ \mathrm{to}\ reduce\ the\ number\ of\ components$$
$$\mathrm{o} = number\ of\ objects$$

Evidence to support the conclusion:

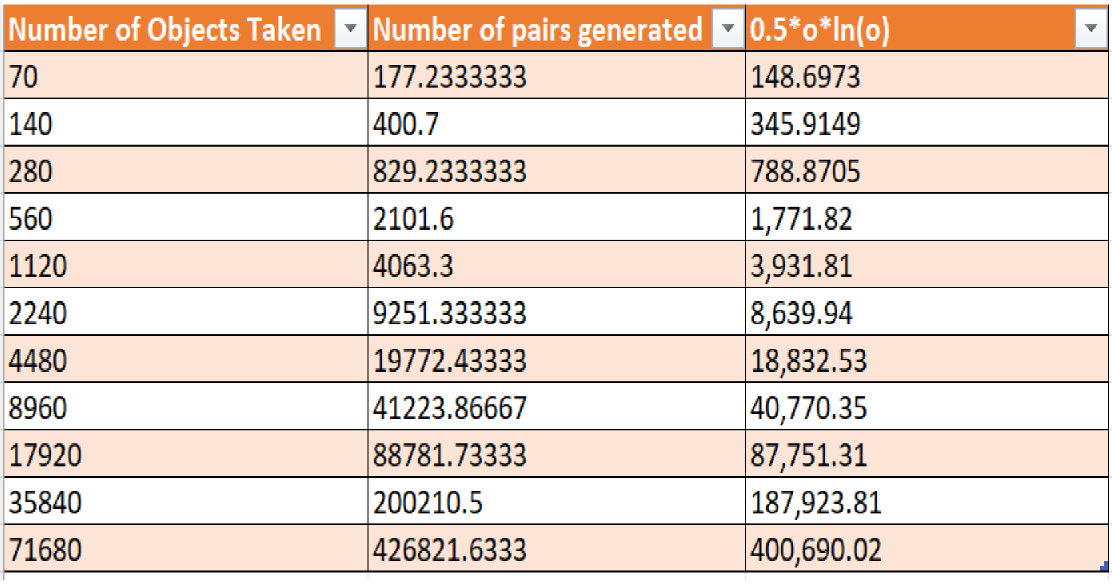
Let $f(\mathrm{o})$ be the number of pairs (p) generated to reduce the number of components from o to 1.

For my case, I am taking the initial value of o to be 70.

We can calculate the number of pairs (p) generated to reduce the number of components from 0 to 1 using the doubling method. Also, the average number of pairs generated to reach our conclusion for each value of p can be computed.

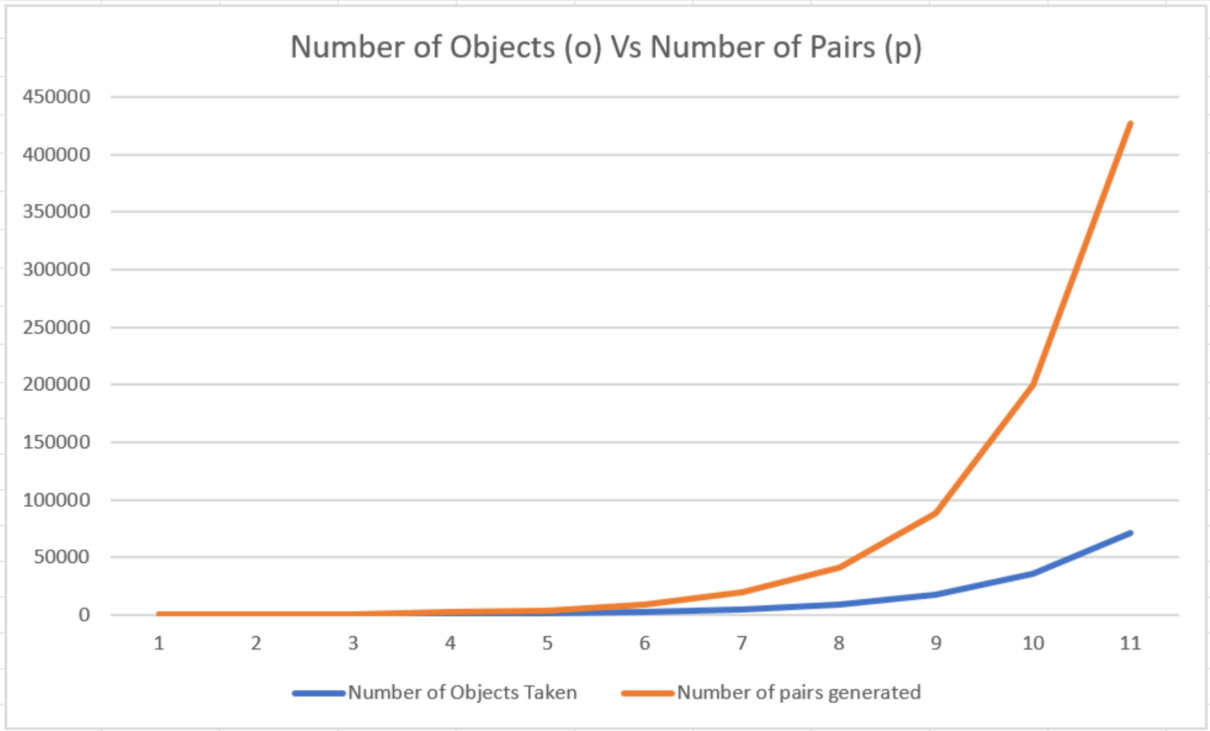
The average number of pairs needed to reduce the component to 1 for larger values of o comes out to be close to ½ $\times\ \mathrm{o} \times \mathrm{lo}(\mathrm{o})$.

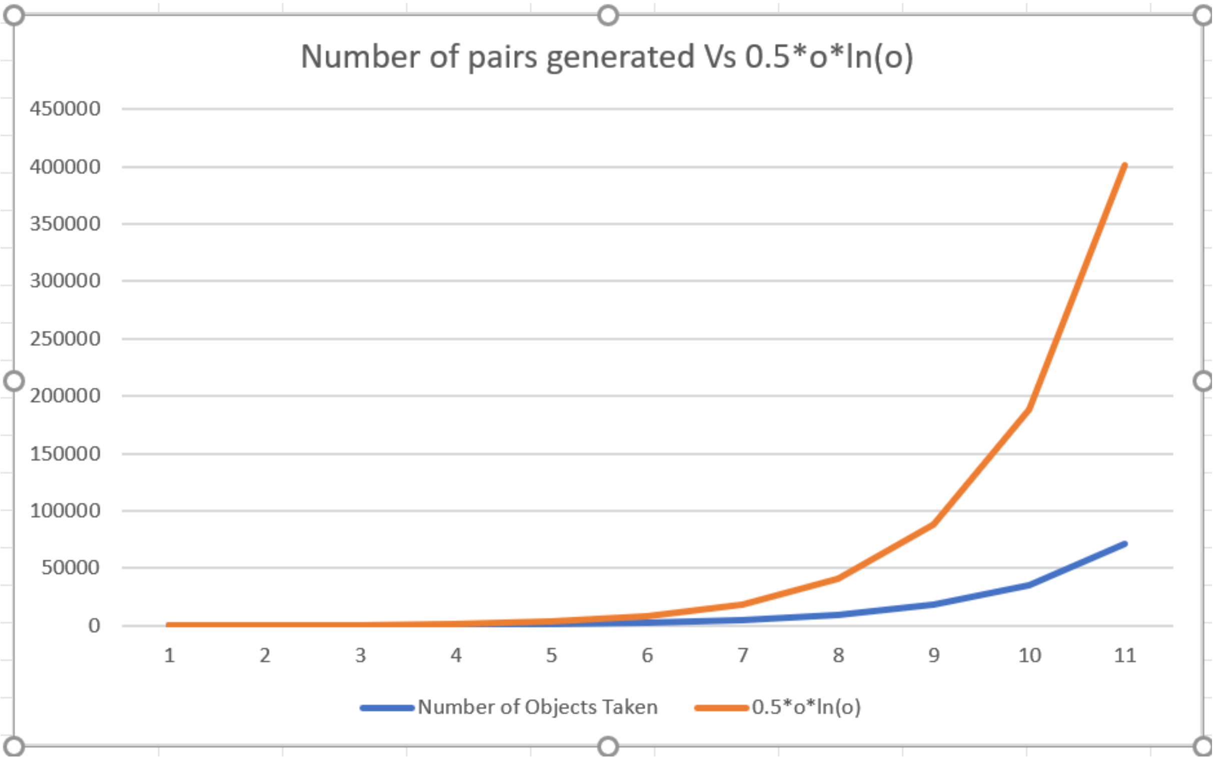
Hence, we can check if the pairs are connected or disconnected $(\mathrm{o}\ lo\ \mathrm{o})$ and only two possibilities come out for each pair, very identical with sorting a list. Therefore the relationship between p and o is almost identical to ½ $\times \mathrm{o} \times \ln(\mathrm{o})$.



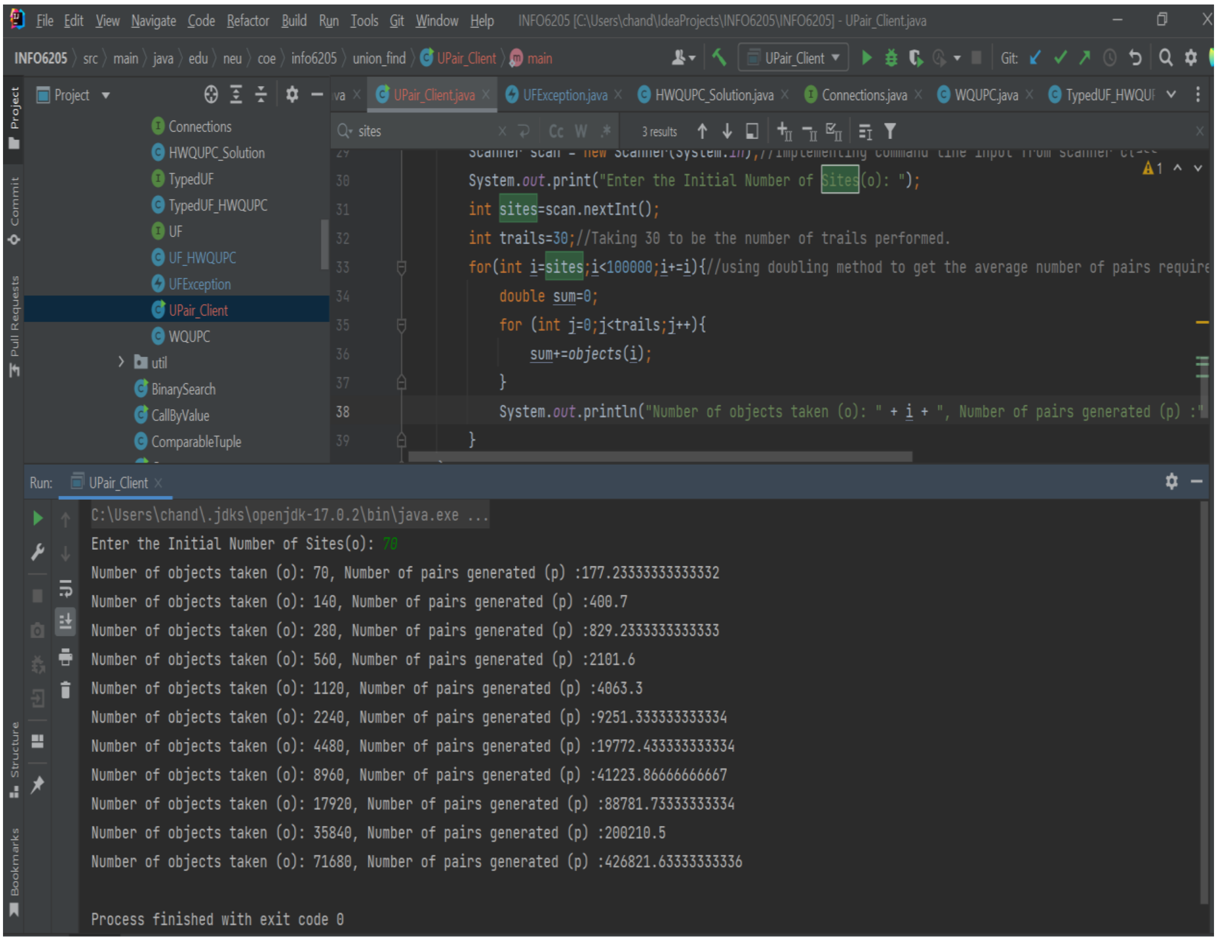
Number of Objects Taken	Number of pairs generated	0.5*o*ln(o)
70	177.2333333	148.6973
140	400.7	345.9149
280	829.2333333	788.8705
560	2101.6	1,771.82
1120	4063.3	3,931.81
2240	9251.333333	8,639.94
4480	19772.43333	18,832.53
8960	41223.86667	40,770.35
17920	88781.73333	87,751.31
35840	200210.5	187,923.81
71680	426821.6333	400,690.02

The below diagrams show the result of plotting the above table data, on a standard scale, with the number of objects (o) on the $x - axis$ and, the number of pairs (p) generated to reduce the number of components from 0 to 1 on the $y - axis$.




Number of pairs generated Vs 0.5*o*ln(o)
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
1
2
3
4
5
6
7
8
9
10
11
Number of Objects Taken
0.5*o*ln(o)

Console Output:



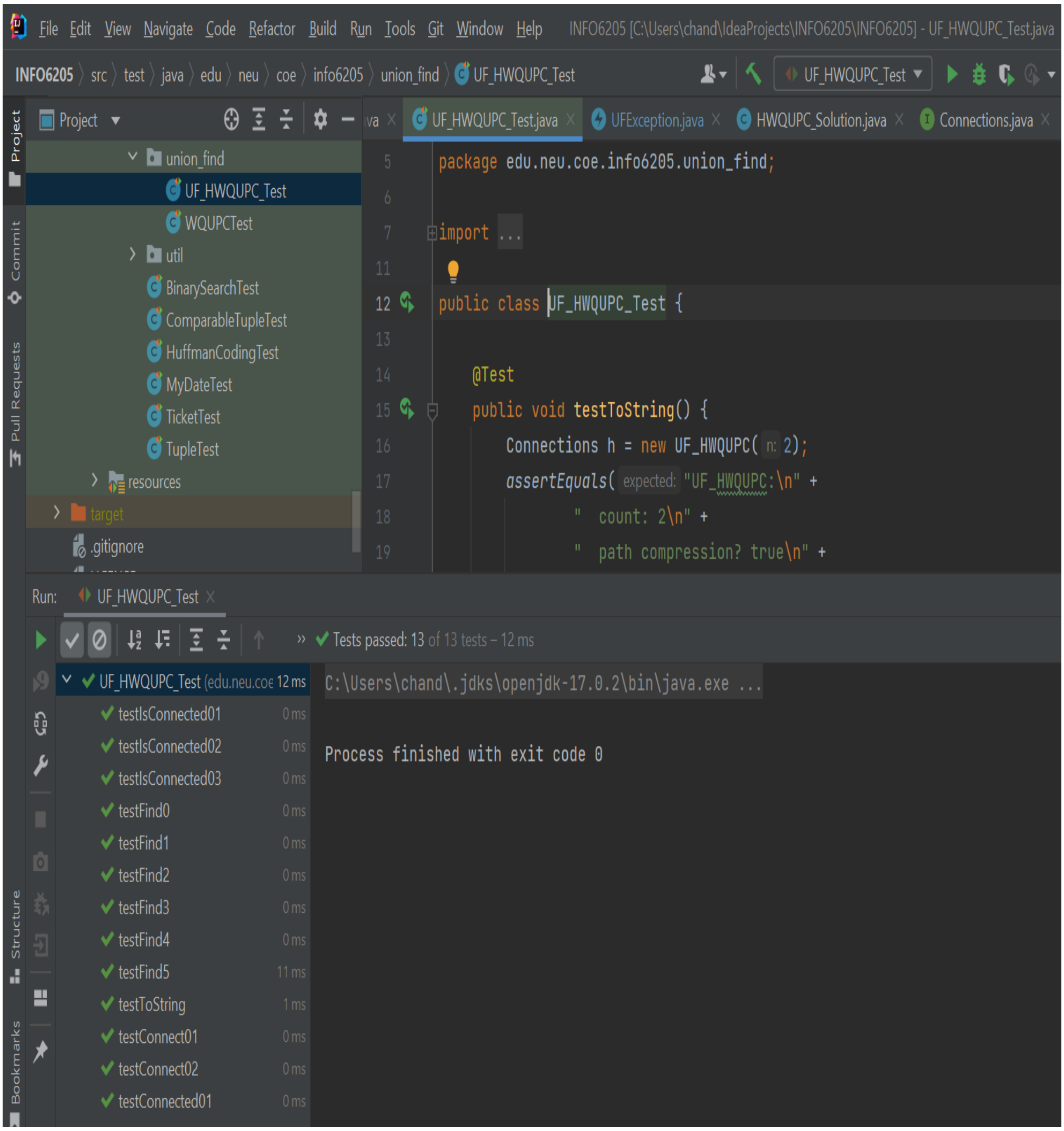
```
        Scanner scan = new Scanner(System.in);//implementing command line input from scanner cl
        System.out.print("Enter the Initial Number of Sites(o): ");
        int sites=scan.nextInt();
        int trails=30;//Taking 30 to be the number of trails performed.
        for(int i=sites;i<100000;i+=i){//using doubling method to get the average number of pairs require
            double sum=0;
            for (int j=0;j<trails;j++){
                sum+=objects(i);
            }
            System.out.println("Number of objects taken (o): " + i + ", Number of pairs generated (p) :"
        }
```

```
C:\Users\chand\.jdks\openjdk-17.0.2\bin\java.exe ...
Enter the Initial Number of Sites(o): 70
Number of objects taken (o): 70, Number of pairs generated (p) :177.23333333333332
Number of objects taken (o): 140, Number of pairs generated (p) :400.7
Number of objects taken (o): 280, Number of pairs generated (p) :829.2333333333333
Number of objects taken (o): 560, Number of pairs generated (p) :2101.6
Number of objects taken (o): 1120, Number of pairs generated (p) :4063.3
Number of objects taken (o): 2240, Number of pairs generated (p) :9251.333333333334
Number of objects taken (o): 4480, Number of pairs generated (p) :19772.433333333334
Number of objects taken (o): 8960, Number of pairs generated (p) :41223.86666666667
Number of objects taken (o): 17920, Number of pairs generated (p) :88781.73333333334
Number of objects taken (o): 35840, Number of pairs generated (p) :200210.5
Number of objects taken (o): 71680, Number of pairs generated (p) :426821.63333333336

Process finished with exit code 0
```

```
Enter the Initial Number of Sites(o): 70
Number of objects taken (o): 70, Number of pairs generated (p) :177.23333333333332
Number of objects taken (o): 140, Number of pairs generated (p) :400.7
Number of objects taken (o): 280, Number of pairs generated (p) :829.2333333333333
Number of objects taken (o): 560, Number of pairs generated (p) :2101.6
Number of objects taken (o): 1120, Number of pairs generated (p) :4063.3
Number of objects taken (o): 2240, Number of pairs generated (p) :9251.333333333334
Number of objects taken (o): 4480, Number of pairs generated (p) :19772.433333333334
Number of objects taken (o): 8960, Number of pairs generated (p) :41223.86666666667
Number of objects taken (o): 17920, Number of pairs generated (p) :88781.73333333334
Number of objects taken (o): 35840, Number of pairs generated (p) :200210.5
Number of objects taken (o): 71680, Number of pairs generated (p) :426821.63333333336

Process finished with exit code 0
```

Unit Test Results:

UF_HWQUPC_Test.java



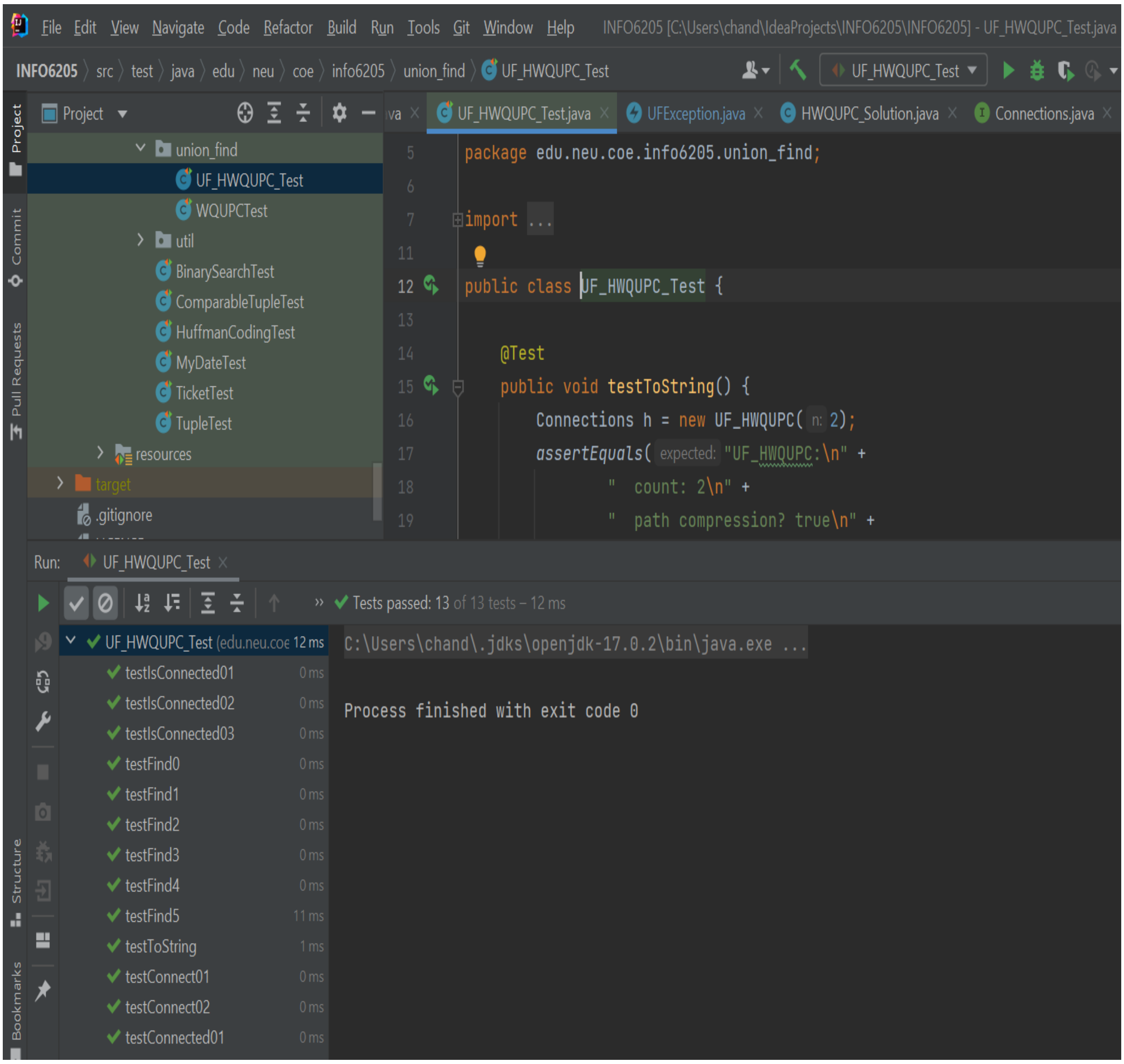
```
package edu.neu.coe.info6205.union_find;

import ...

public class UF_HWQUPC_Test {

    @Test
    public void testToString() {
        Connections h = new UF_HWQUPC( n: 2);
        assertEquals( expected: "UF_HWQUPC:\n" +
                "  count: 2\n" +
                "  path compression? true\n" +
```

✓ Tests passed: 13 of 13 tests – 12 ms

```
✓ UF_HWQUPC_Test (edu.neu.coe 12 ms
    ✓ testIsConnected01   0 ms
    ✓ testIsConnected02   0 ms
    ✓ testIsConnected03   0 ms
    ✓ testFind0           0 ms
    ✓ testFind1           0 ms
    ✓ testFind2           0 ms
    ✓ testFind3           0 ms
    ✓ testFind4           0 ms
    ✓ testFind5          11 ms
    ✓ testToString        1 ms
    ✓ testConnect01       0 ms
    ✓ testConnect02       0 ms
    ✓ testConnected01     0 ms
```

```
C:\Users\chand\.jdks\openjdk-17.0.2\bin\java.exe ...

Process finished with exit code 0
```

WQUPC_Test.java



```java
package edu.neu.coe.info6205.union_find;

import ...

public class UF_HWQUPC_Test {

    @Test
    public void testToString() {
        Connections h = new UF_HWQUPC( n: 2);
        assertEquals( expected: "UF_HWQUPC:\n" +
                "  count: 2\n" +
                "  path compression? true\n" +
```

✓ Tests passed: 13 of 13 tests – 12 ms

```
✓ UF_HWQUPC_Test (edu.neu.coe 12 ms
    ✓ testIsConnected01     0 ms
    ✓ testIsConnected02     0 ms
    ✓ testIsConnected03     0 ms
    ✓ testFind0             0 ms
    ✓ testFind1             0 ms
    ✓ testFind2             0 ms
    ✓ testFind3             0 ms
    ✓ testFind4             0 ms
    ✓ testFind5            11 ms
    ✓ testToString          1 ms
    ✓ testConnect01         0 ms
    ✓ testConnect02         0 ms
    ✓ testConnected01       0 ms
```

```
C:\Users\chand\.jdks\openjdk-17.0.2\bin\java.exe ...

Process finished with exit code 0
```