

Assignment-3

Vennela Reddy Karaddi
NU ID : 001081643

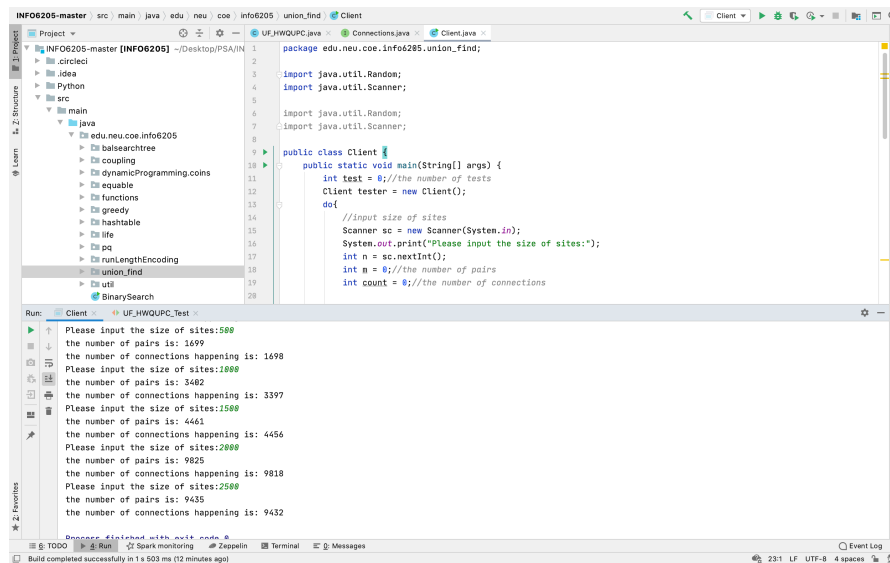
Problem:

Implement height-weighted Quick Union with Path Compression. Check that the unit tests for this class all work.

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 $n-1$, determine if they are connected and `union()` if not. 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.

Determine the relationship between the number of objects (n) and the number of pairs (m) generated.

Output:



The screenshot shows an IDE with a project named 'INFO6205-master'. The 'src' directory contains a 'main' package with a 'edu.neu.coe.info6205' sub-package. The 'edu.neu.coe.info6205' package contains several classes, including 'union_find'. The 'Client.java' file is open, showing the following code:

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

import java.util.Random;
import java.util.Scanner;

import java.util.Random;
import java.util.Scanner;

public class Client {
    public static void main(String[] args) {
        int test = 0; // the number of tests
        Client tester = new Client();
        do {
            // input size of sites
            Scanner sc = new Scanner(System.in);
            System.out.print("Please input the size of sites:");
            int n = sc.nextInt();
            int m = 0; // the number of pairs
            int count = 0; // the number of connections
        } while (true);
    }
}
```

The 'Run' window shows the output of the 'Client' program, which prompts the user to input the size of sites and the number of pairs, and then prints the number of connections happening. The output is as follows:

```
Please input the size of sites:500
the number of pairs is: 1699
the number of connections happening is: 1698
Please input the size of sites:1000
the number of pairs is: 3402
the number of connections happening is: 3397
Please input the size of sites:1500
the number of pairs is: 4461
the number of connections happening is: 4456
Please input the size of sites:2000
the number of pairs is: 9825
the number of connections happening is: 9818
Please input the size of sites:2500
the number of pairs is: 9435
the number of connections happening is: 9432
```

Testcases:

The screenshot shows an IDE with a project named 'INFO6205-master'. The 'test' directory contains a 'java' subdirectory with a package 'edu.neu.coe.info6205'. Inside this package is a class 'UF_HWQUPC_Test'. The code for 'UF_HWQUPC_Test' is as follows:

```
1 //...
2
3 package edu.neu.coe.info6205.union_find;
4
5 import ...
6
7 public class UF_HWQUPC_Test {
8
9     @Test
10     public void testToString() {
11         Connections h = new UF_HWQUPC(1000);
12         assertEquals("expected: \"UF_HWQUPC:\\n\" +
13             \"count: 2\\n\" +
14             \"path compression? true\\n\" +
15             \"parents: [0, 1]\\n\" +
16             \"heights: [1, 1]\", h.toString());
17     }
18 }
19
20 /**
21  *
22  */
23
24 //...
25
```

The 'Run' tab shows the test results for 'UF_HWQUPC_Test'. The tests passed are:

- testisConnected01 (1 ms)
- testisConnected02 (0 ms)
- testisConnected03 (2 ms)
- testFind0 (0 ms)
- testFind1 (0 ms)
- testFind2 (0 ms)
- testFind3 (1 ms)
- testFind4 (0 ms)
- testFind5 (1 ms)
- testToString (0 ms)
- testConnect01 (0 ms)
- testConnect02 (0 ms)
- testConnect03 (0 ms)

Tests passed: 13 of 13 tests - 5 ms

Conclusion :

As number of objects increases, number of pairs also increase in $m = n/2 \log n$;

When $n = 500$; $m = 500/2 \log 500 = 2225$

When $n = 1000$; $m = 1000/2 \log 1000 = 4980$

When $n = 2500$; $m = 2500/2 \log 2500 = 13750$

Below values are from my program which are close to relation above.

number of objects	Number of pairs generated
500	1699
1000	3402
1500	4461
2000	9825
2500	9435

