Program Structures and Algorithms
Spring 2023(SEC –8)

NAME: Junlong Qiao
NUID: 002784609

Task: Assignment 4(WQUPC)

Implement height-weighted Quick Union with Path Compression and check that the unit tests for this class all work. Develop a UF client to find the relationship between the number of objects and the pairs

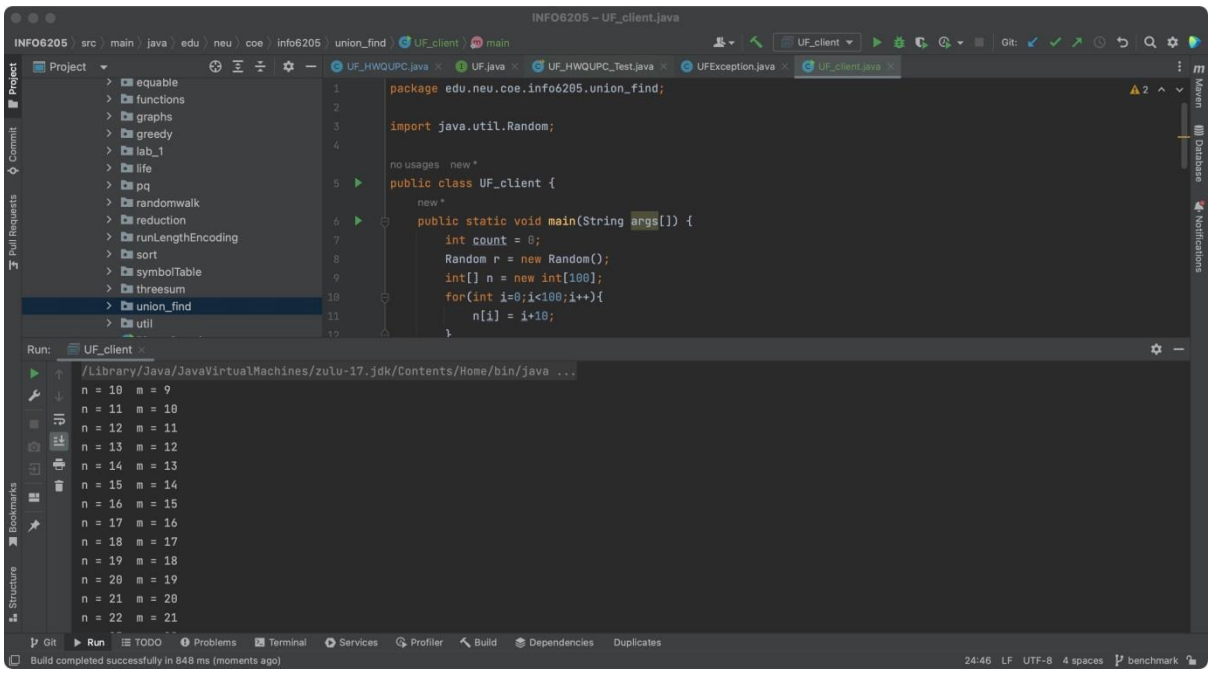
Runtie Relationship Conclusion:

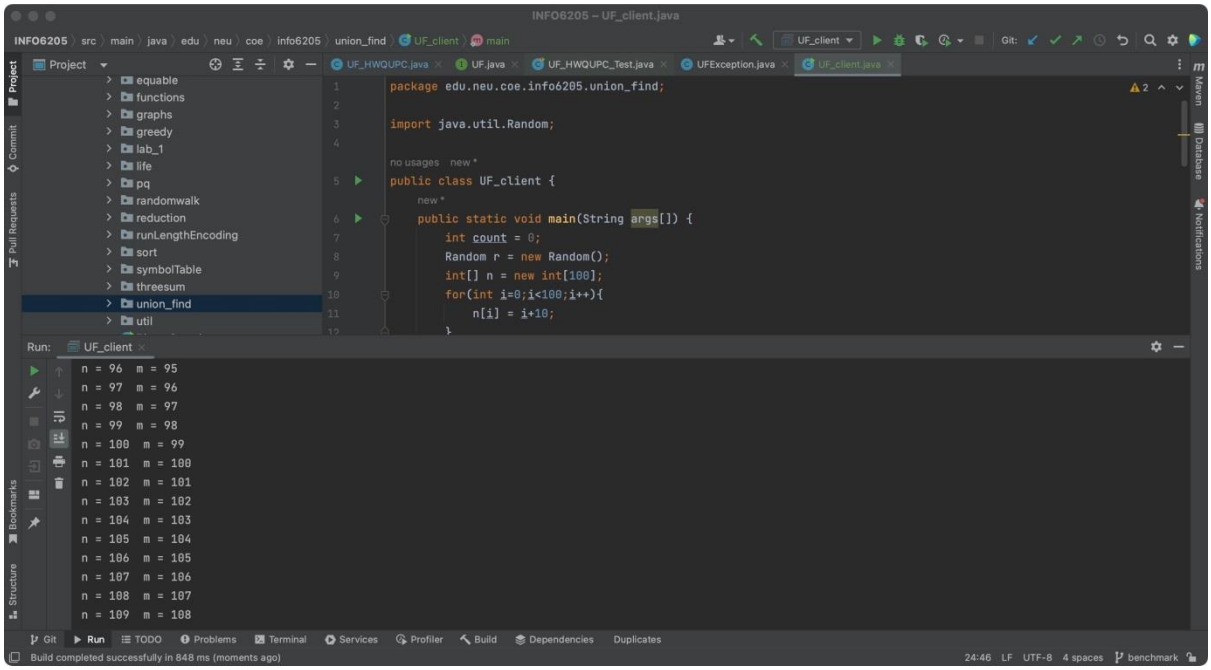
At first I thought that m grew linearly with n, but as n increased, the rate of increase of m was accelerating and accelerating very slowly, not up to the quadratic power of n, so I speculated m = k*n*lgn. After fitting, the relationship is obtained as follows

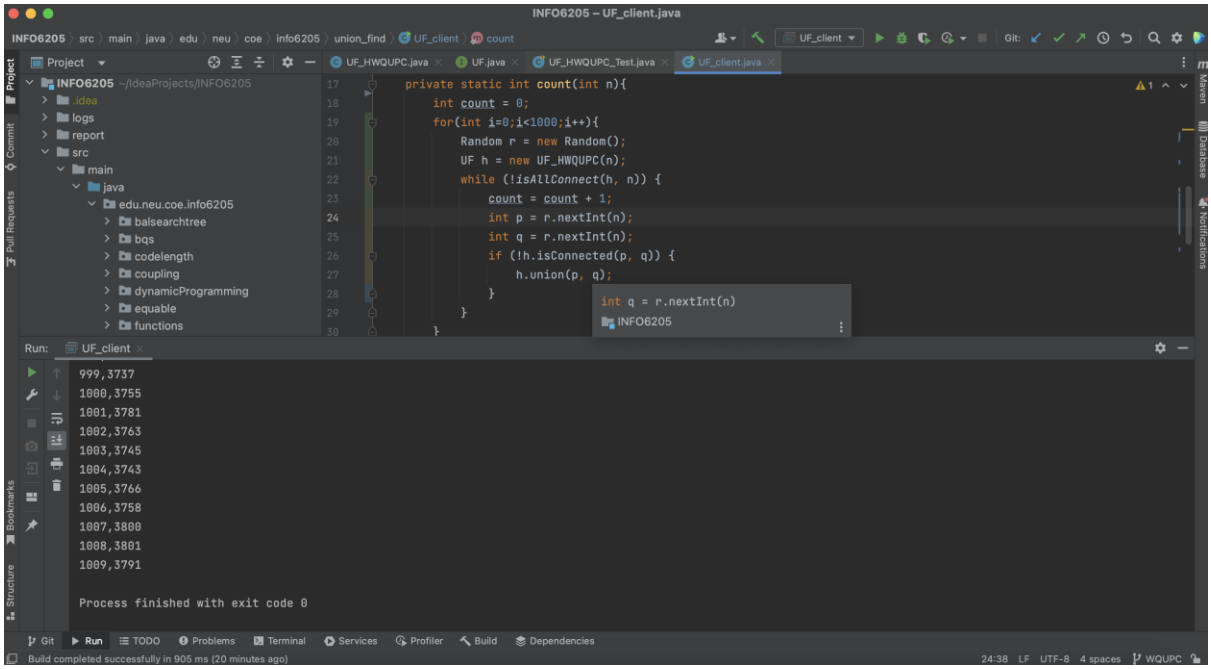
$$\mathbf{m = 0.54 * n * lgn}$$

Evidence to support that conclusion:

The average value of m is obtained by running n from 10 to 1009 each for 1000 times as follows



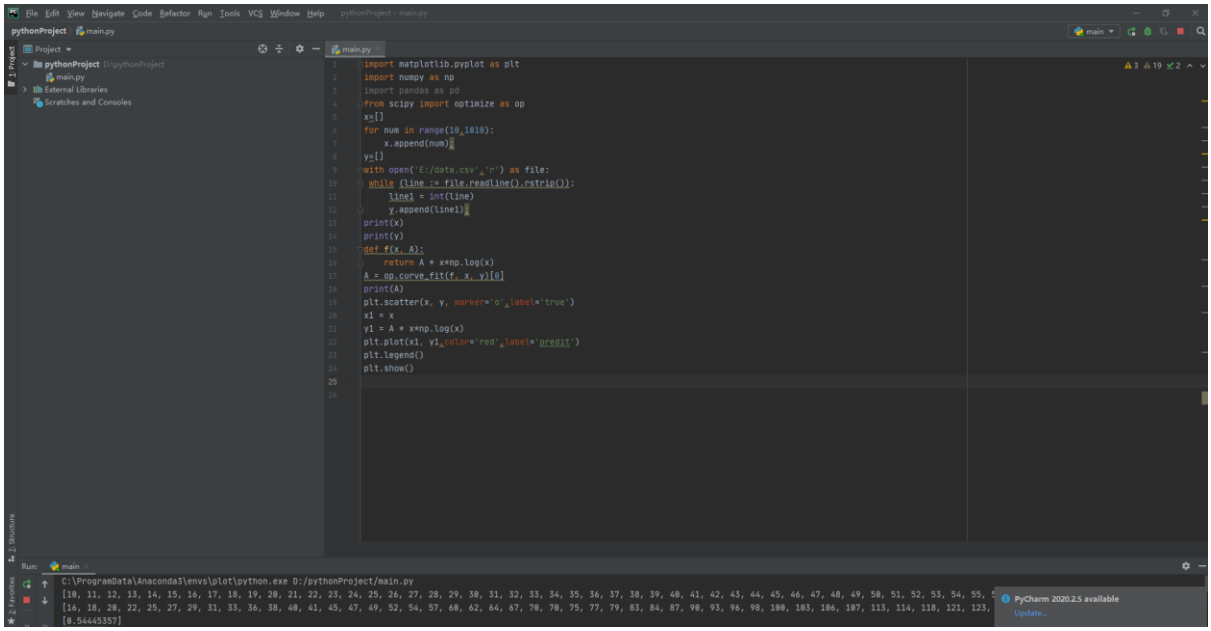


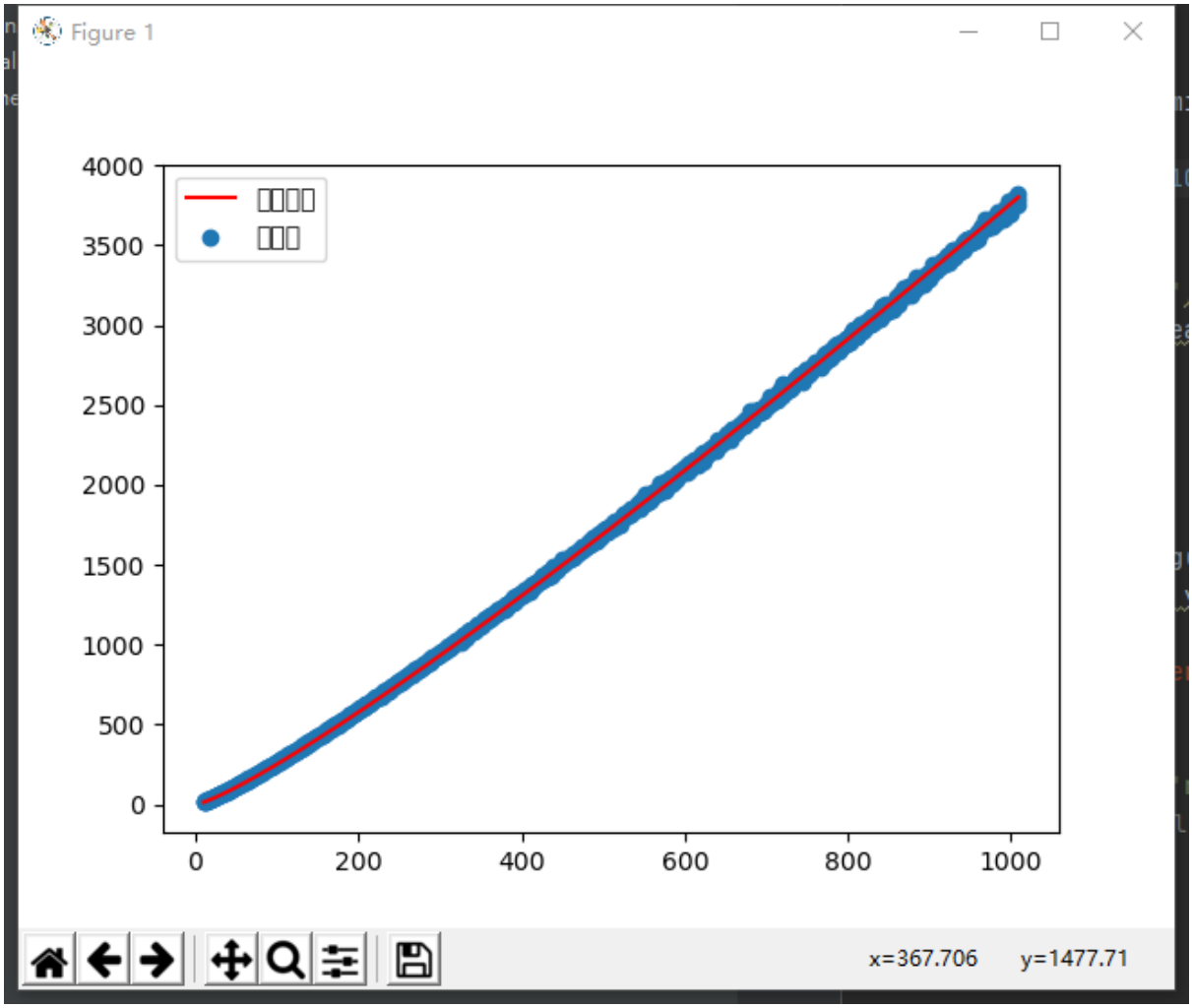


n	m
10	16
11	18
12	20
13	22
14	24
15	27

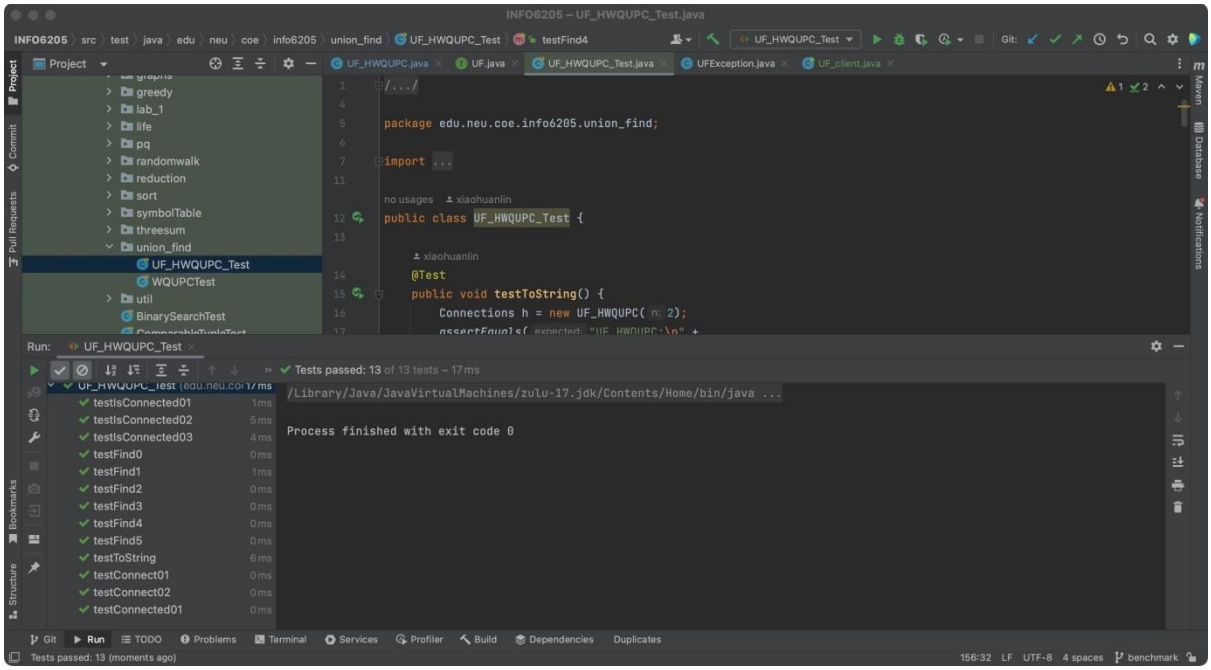
16	29
17	30
18	33
19	35
…	…
1009	3817

Use python to fit with k*n*logn as the target.





Unit Test Screenshots:



Code of 3-Sum:

UF-client.java

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

import java.util.Random;

public class UF_client {
    public static void main(String args[]) {
        int[] n = new int[1000];
        for(int i=0;i<1000;i++){
            n[i] = i+10;
        }
        for (int i = 0; i < n.length; i++) {
            int m  = count(n[i]);
            System.out.println((i+10) +","+m);
        }
    }

    private static int count(int n){
        int count = 0;
        for(int i=0;i<1000;i++){
            Random r = new Random();
            UF h = new UF_HWQUPC(n);
            while (!isAllConnect(h, n)) {
                count = count + 1;
                int p = r.nextInt(n);
                int q = r.nextInt(n);
                if (!h.isConnected(p, q)) {
                    h.union(p, q);
                }
            }
        }
        return count/1000;
    }
    private static boolean isAllConnect(UF h, int n){
        for(int i=1;i<n;i++){
            if(!h.isConnected(0,i)){
                return false;
            }
        }
        return true;
    }
}
```