Shrusti Rajesh Chheda

NUID : 002196756

Program Structures and Algorithms

Fall 2021

Assignment No. 3 – WQUPC

- Tasks in the assignment:

Part 1: Height-weighted Quick Union with Path Compression

1. Implemented height weighted quick union where we maintain an array to keep track of the tree height and while union add the shorter tree to the taller tree.
2. Path compression is done using single pass path-halving mechanism where we update root to that of the grandparent. This reduces the time taken by find().
3. Once a site is found we check if they're connected else union is performed.
4. Merge the components which contain site p with site q using strategy mentioned in point 1.

Part 2: Union Find Client

1. Developed a union find client which contains a main() method and a count() method.
2. Takes number of sites from the command line and number of times we wish to double the sites.
3. For each random pair of integer generated, a connection is established if it's not already present. The output is the number of connections till all sites are connected under a single component.
4. I've performed this operation 100 times for each site and taken an average of the connections generated.

Part 3: Conclusions and observations

1. The relationship between the number of objects (*n*) and the number of pairs (*m*) generated till all sites are connected.

2. Observations are mentioned below.

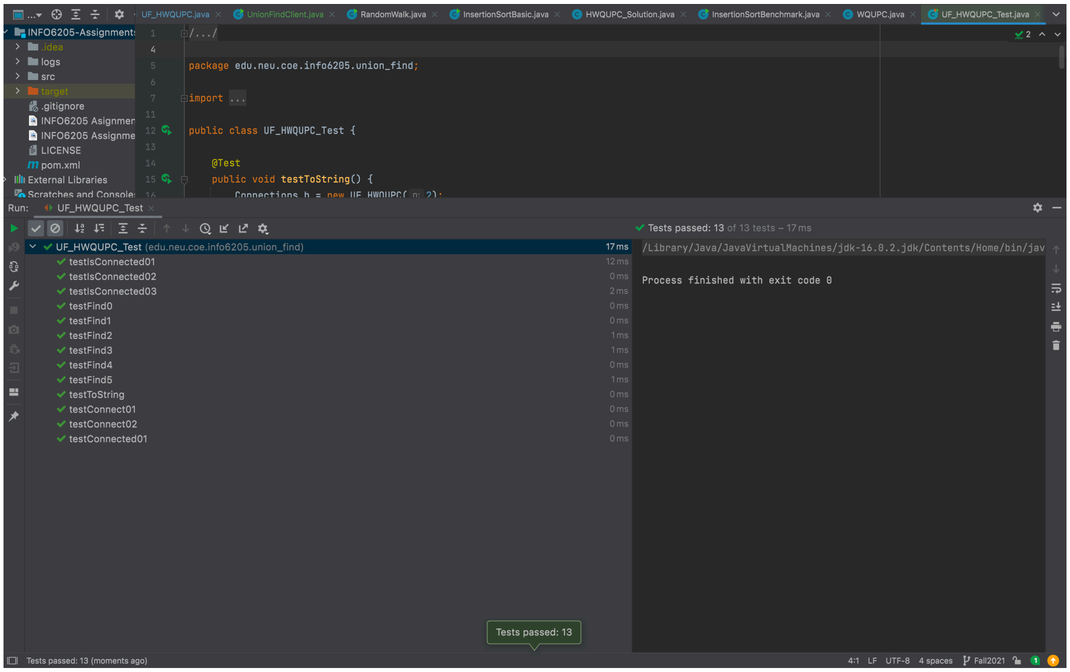
- Relationship Conclusion

Relationship between 'n' ie the number of sites taken and 'm' ie. the count of pairs generated is :

m=n(logn)/2

- Evidence to support conclusions

1. Snapshot of unit test part 1



```
/.../

package edu.neu.coe.info6205.union_find;

import ...

public class UF_HWQUPC_Test {

    @Test
    public void testToString() {
        Connections h = new UF_HWQUPC( n: 2);
```

```
UF_HWQUPC_Test (edu.neu.coe.info6205.union_find)   17 ms
    testIsConnected01   12 ms
    testIsConnected02    0 ms
    testIsConnected03    2 ms
    testFind0            0 ms
    testFind1            0 ms
    testFind2            1 ms
    testFind3            1 ms
    testFind4            0 ms
    testFind5            1 ms
    testToString         0 ms
    testConnect01        0 ms
    testConnect02        0 ms
    testConnected01      0 ms
```

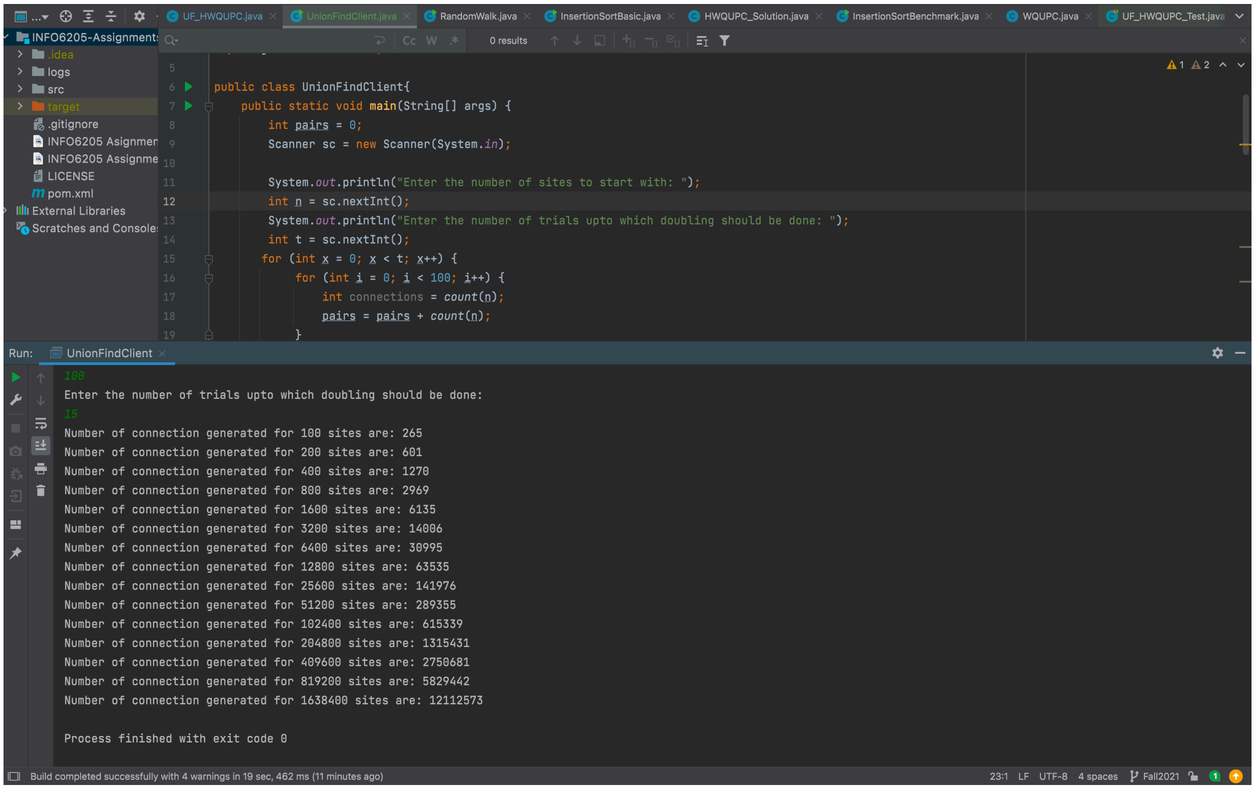
Tests passed: 13 of 13 tests – 17 ms

```
/Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/jav

Process finished with exit code 0
```

Tests passed: 13

Tests passed: 13 (moments ago)

2. Snapshot of output part 2



```
public class UnionFindClient{
    public static void main(String[] args) {
        int pairs = 0;
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number of sites to start with: ");
        int n = sc.nextInt();
        System.out.println("Enter the number of trials upto which doubling should be done: ");
        int t = sc.nextInt();
       for (int x = 0; x < t; x++) {
            for (int i = 0; i < 100; i++) {
                int connections = count(n);
                pairs = pairs + count(n);
            }
```

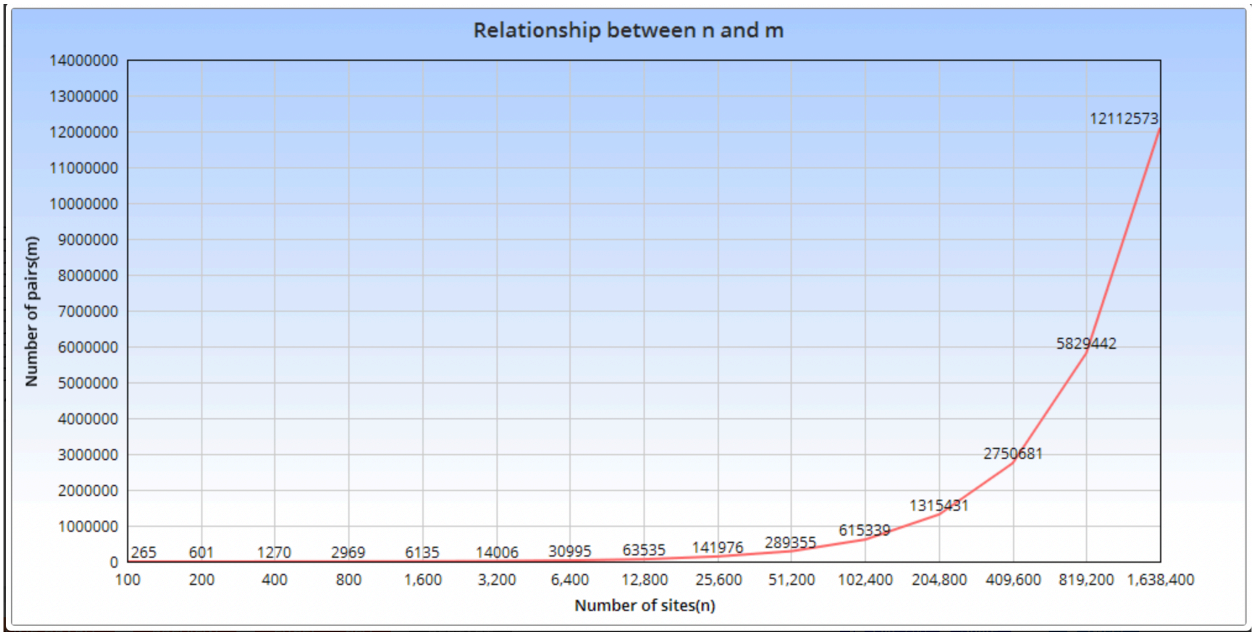
```
100
Enter the number of trials upto which doubling should be done:
15
Number of connection generated for 100 sites are: 265
Number of connection generated for 200 sites are: 601
Number of connection generated for 400 sites are: 1270
Number of connection generated for 800 sites are: 2969
Number of connection generated for 1600 sites are: 6135
Number of connection generated for 3200 sites are: 14006
Number of connection generated for 6400 sites are: 30995
Number of connection generated for 12800 sites are: 63535
Number of connection generated for 25600 sites are: 141976
Number of connection generated for 51200 sites are: 289355
Number of connection generated for 102400 sites are: 615339
Number of connection generated for 204800 sites are: 1315431
Number of connection generated for 409600 sites are: 2750681
Number of connection generated for 819200 sites are: 5829442
Number of connection generated for 1638400 sites are: 12112573

Process finished with exit code 0
```

Build completed successfully with 4 warnings in 19 sec, 462 ms (11 minutes ago)

Observations:

Relationship between n and m

Number of sites(n)	Number of pairs(m) ≈ n(logn)/2	Value of n(logn)
100	265	460
200	601	1,059
400	1,270	2,397
800	2,969	5,348
1,600	6,135	11,804
3,200	14,006	25,827
6,400	30,995	56,090
12,800	63,535	121,052
25,600	141,976	259,849
51,200	289,355	555,187
102,400	615,339	1,181,352
204,800	1,315,431	2,504,661
409,600	2,750,681	5,293,235
819,200	5,829,442	11,154,295
1,638,400	12,112,573	23,444,243



Conclusion:

Weighted quick union makes sure length of the tree is not too long. By storing the height of an array we reduce the the time that find() takes. We also reduce the number of array access thus reducing the number of random pairs generated from MN, where M is the number of connections and N is the number of sites, to M(logN) approximately . In worst case WQCPC will have height of tree to be logN.

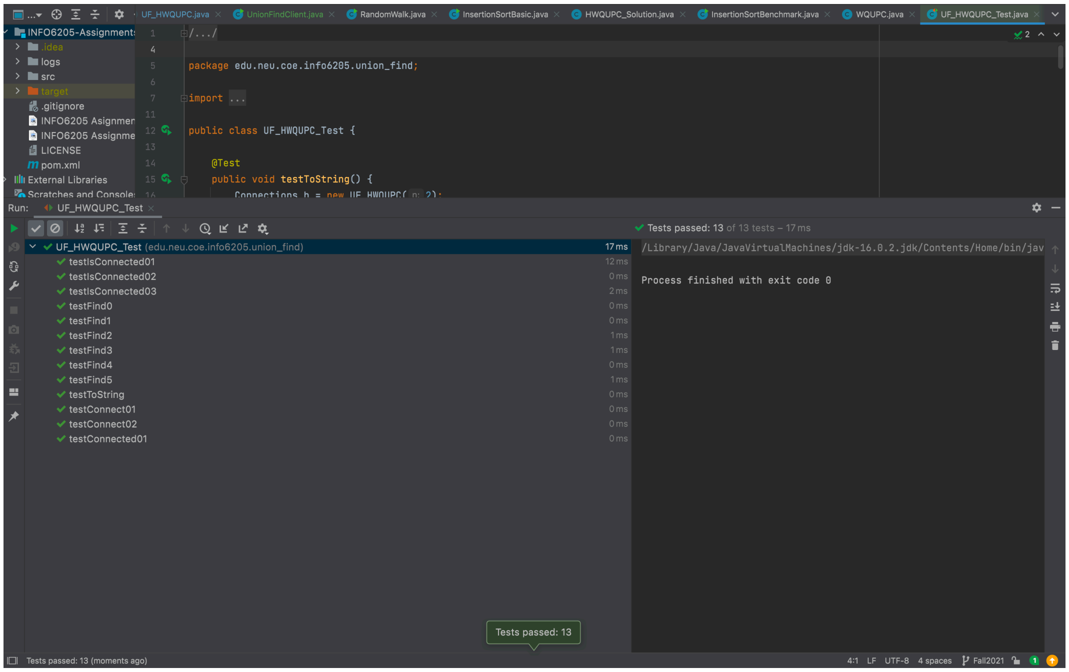
From the graph we can infer that the relationship between M and N is logarithmic . The path compression by halving will reduce the time taken by find() operation reducing the height by 0.5

Hence from graph observations and proof we can conclude the relationship between number of sites(n) and number of connections generated(m) is:

$$m= n(\log(n))/2$$

- Unit test results:

All test cases passed successfully .



```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);
```

```
UF_HWQUPC_Test (edu.neu.coe.info6205.union_find)  17 ms
    testIsConnected01                             12 ms
    testIsConnected02                              0 ms
    testIsConnected03                              2 ms
    testFind0                                      0 ms
    testFind1                                      0 ms
    testFind2                                      1 ms
    testFind3                                      1 ms
    testFind4                                      0 ms
    testFind5                                      1 ms
    testToString                                   0 ms
    testConnect01                                  0 ms
    testConnect02                                  0 ms
    testConnected01                                0 ms
```

```
Tests passed: 13 of 13 tests – 17 ms

/Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/jav

Process finished with exit code 0
```

Tests passed: 13 (moments ago)