Aishwarya Balyaya

001586556

Program Structure & Algorithms

Spring 2022

Assignment 3

Task: To implement height-weighted Quick union with Path compression.

For this task, UF_HWQUPC java class was used and following methods were implemented :

- **find ():** to update the root of input object if path compression is performed
- **mergeComponents():** to merge 2 subtrees such that smaller root points to larger root
- **doPathCompression():** a method that implements the single-pass process of path compression method.

Executed the **Union Find code for height-weighted quick union**. Summarized the relationship between a **number of random pairs generated** to reduce the number of components from **n to 1** for various different components of n.

To perform and test the implementation of UF_HWQUPC class, the UnionFind java class was created

union_find 〉 UF_HWQUPC 〉 doPathCompression

WQUPC.java | UF_HWQUPC.java | UF_HWQUPC_Test.java | UnionFind.java

```java
     * @return the number of components (between {@code 1} and
     */
    public int components() { return count; }

    /**
     * Returns the component identifier for the component conta
     *
     * @param p the integer representing one site
     * @return the component identifier for the component conta
     * @throws IllegalArgumentException unless {@code 0 <= p <
     */
    public int find(int p) {
        validate(p);
        int root = p;
        while(root != parent[root]){
            root = parent[root];
        }
        if(pathCompression){
            doPathCompression(p);
        }
        return root;
    }
```

union_find 〉 UF_HWQUPC 〉 doPathCompression

WQUPC.java | UF_HWQUPC.java | UF_HWQUPC_Test.java | UnionFind.java

```java
    private final int[] parent;   // parent[i] = parent of i
    private final int[] height;   // height[i] = height of subtree rooted at i
    private int count;  // number of components
    private boolean pathCompression;

    private void mergeComponents(int i, int j) {
        // FIXME make shorter root point to taller one
        // END
        if (i==j) return;
        if (height[i] < height[j]) {
            parent[i]=j;
            height[j] += height[i];
        }
        else {
            parent[j] = i;
            height[i] += height[j];
        }
    }

    /**
     * This implements the single-pass path-halving mechanism of path compression
     */
    private void doPathCompression(int i) {
        // FIXME update parent to value of grandparent
        // END
        while(i != parent[i]){
            parent[i] = parent[parent[i]];
            i = parent[i];
        }
    }
}
```

Console Output:

n:8 Total connections generated: 7

n:16 Total connections generated: 42

n:32 Total connections generated: 83

n:64 Total connections generated: 179

n:128 Total connections generated: 230

n:256 Total connections generated: 666

n:512 Total connections generated: 1935

n:1024 Total connections generated: 4655

n:2048 Total connections generated: 9103

n:4096 Total connections generated: 18000

n:8192 Total connections generated: 40339

n:16384 Total connections generated: 83646

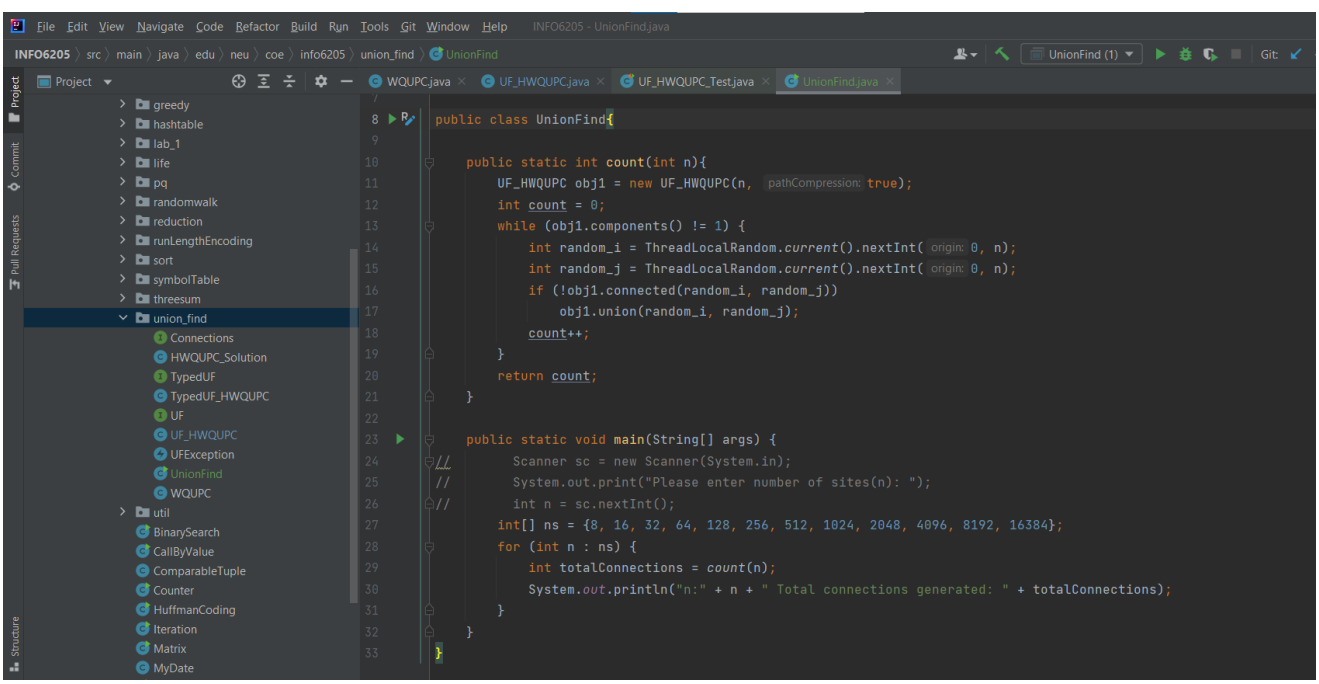
Relationship:

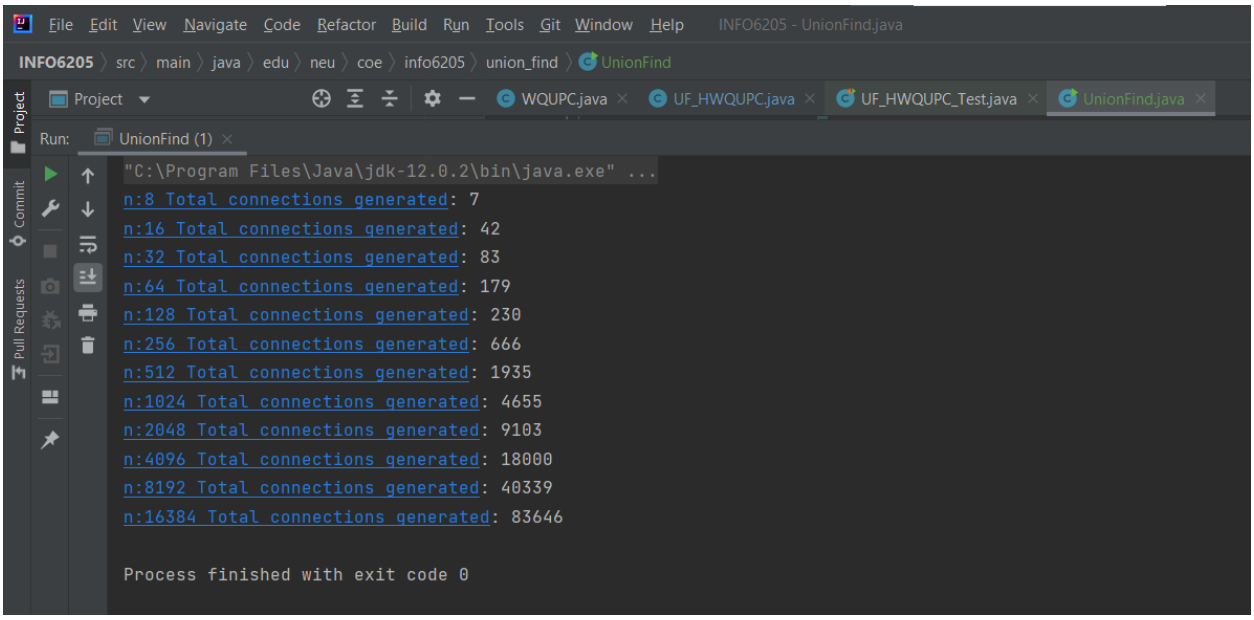
From the results stated above, It can be concluded that to reduce the number of components from **n to 1,** the relationship between **Number of Objects** and the **Number of Pairs** generated is N/2 * log(N).

i.e. For a number of **32 components** in Quick Union, approximately **55 random pairs** are generated to accomplish the singular component condition.

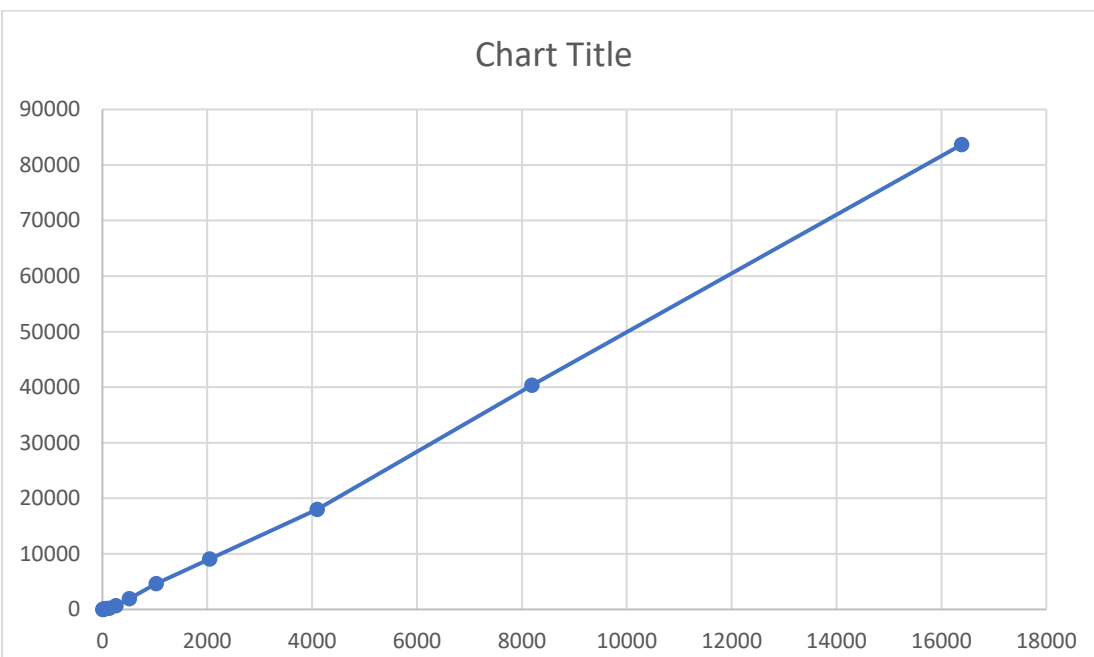
Evidence:

I have attached a table and a chart to show the relationship between the number of objects and a number of pairs generated

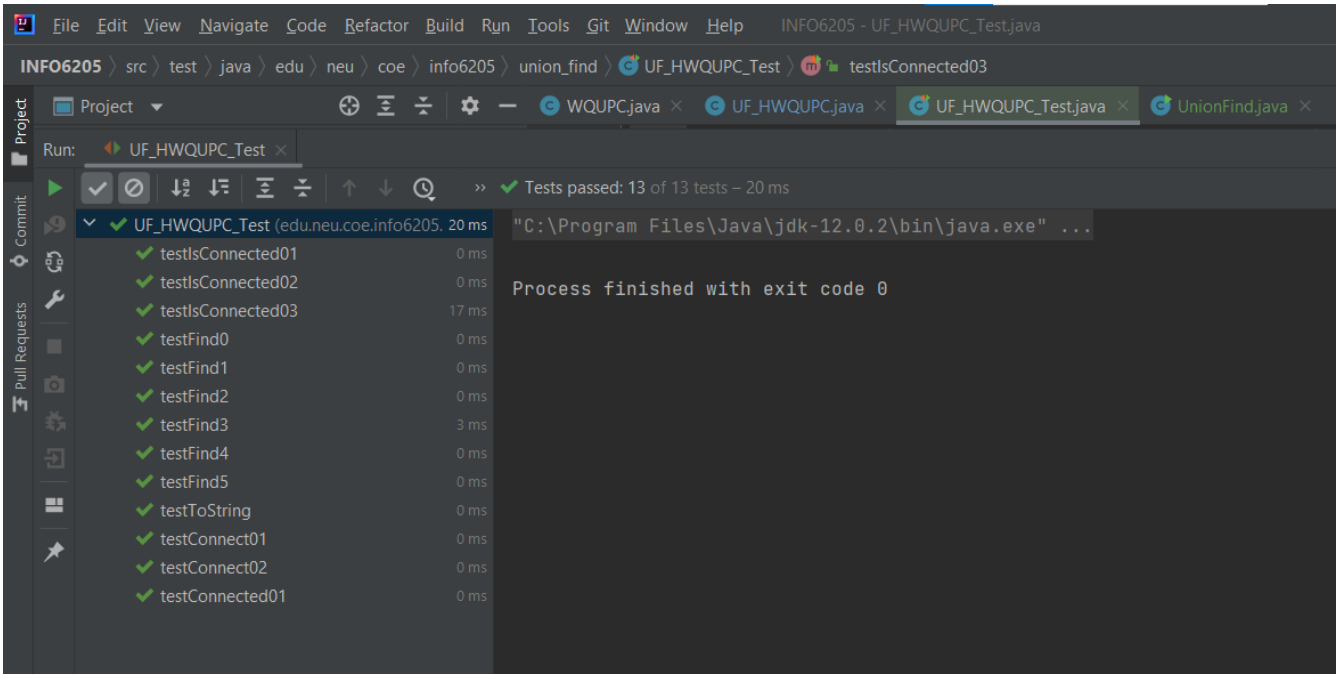




Graphical Representation: N vs N/2 ln(N)



Unit Test:


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
INFO6205 - UF_HWQUPC_Test.java
INFO6205 src test java edu neu coe info6205 union_find UF_HWQUPC_Test testIsConnected03
Project
WQUPC.java
UF_HWQUPC.java
UF_HWQUPC_Test.java
UnionFind.java
Run: UF_HWQUPC_Test
Tests passed: 13 of 13 tests – 20 ms
UF_HWQUPC_Test (edu.neu.coe.info6205. 20 ms
testIsConnected01 0 ms
testIsConnected02 0 ms
testIsConnected03 17 ms
testFind0 0 ms
testFind1 0 ms
testFind2 0 ms
testFind3 3 ms
testFind4 0 ms
testFind5 0 ms
testToString 0 ms
testConnect01 0 ms
testConnect02 0 ms
testConnected01 0 ms
"C:\Program Files\Java\jdk-12.0.2\bin\java.exe" ...
Process finished with exit code 0