

**SANTHOSH KRISHNA MADDI**

**NUID : 002127881**

## **INFO 6205 - ASSIGNMENT 3**

### **Task:**

- To implement height-weighted Quick Union with Path Compression and check if all the unit tests for this class work.
- Using the implementation of UF\_HWQUPC, develop a UF client that takes an integer value  $n$  from the command line and generates random pairs to connect until it has a single component.
- To determine the relation between the number of objects  $n$  and number of random pairs generated  $m$ .

### **Relation:**

After series of experiments, by passing 5 different values of  $n$  and calculating the average from for five different runs of each value, we can conclude that the relationship between number of objects  $n$  and number of connections  $m$  can be inferred as:

$$\mathbf{M \approx 1/2 N \log N}$$

## **Findings:**

### **Running 5 times for $n = 1000$**

- 1.Total number of pairs generated for 1000 objects are: 3305
- 2.Total number of pairs generated for 1000 objects are: 3936
- 3.Total number of pairs generated for 1000 objects are: 4634
- 4.Total number of pairs generated for 1000 objects are: 3427
- 5.Total number of pairs generated for 1000 objects are: 3126

Average Value of number of pairs for  $n = 1000$  is: 3685

### **Running 5 times for $n = 2000$**

- 1.Total number of pairs generated for 2000 objects are: 8841
- 2.Total number of pairs generated for 2000 objects are: 7933
- 3.Total number of pairs generated for 2000 objects are: 9454
- 4.Total number of pairs generated for 2000 objects are: 10812
- 5.Total number of pairs generated for 2000 objects are: 7708

Average Value of number of pairs for  $n = 2000$  is: 8949

### **Running 5 times for $n = 3000$**

- 1.Total number of pairs generated for 3000 objects are: 12836
- 2.Total number of pairs generated for 3000 objects are: 13651

3.Total number of pairs generated for 3000 objects are: 11387

4.Total number of pairs generated for 3000 objects are: 13708

5.Total number of pairs generated for 3000 objects are: 10800

Average Value of number of pairs for  $n = 3000$  is: 12476

### **Running 5 times for $n = 4000$**

1.Total number of pairs generated for 4000 objects are: 14610

2.Total number of pairs generated for 4000 objects are: 17226

3.Total number of pairs generated for 4000 objects are: 18565

4.Total number of pairs generated for 4000 objects are: 19814

5.Total number of pairs generated for 4000 objects are: 19160

Average Value of number of pairs for  $n = 4000$  is: 17875

### **Running 5 times for $n = 5000$**

1.Total number of pairs generated for 5000 objects are: 20936

2.Total number of pairs generated for 5000 objects are: 24936

3.Total number of pairs generated for 5000 objects are: 22976

4.Total number of pairs generated for 5000 objects are: 19258

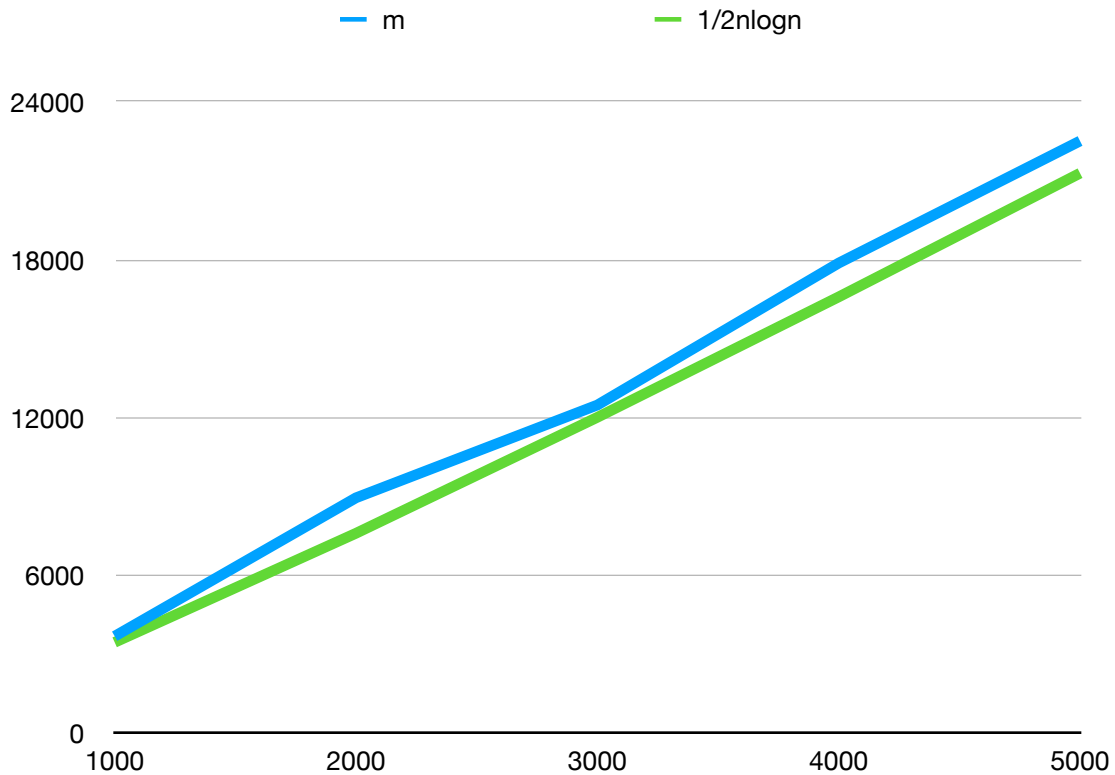
5.Total number of pairs generated for 5000 objects are: 24483

Average Value of number of pairs for  $n = 5000$  is: 22517

The above average values for n and m along with  $1/2n\log n$  are in the following table. (Log base 2)

N	M	$1/2 N\log N$
1000	3685	3453
2000	8949	7600
3000	12476	12009
4000	17875	16588
5000	22517	21292

The above table values are plotted as shown below to determine relationship between n and m.



From the above graph it can be concluded that  $m$  is approximately equal to  $1/2*n\log n$ . (base 2)

## Output Screenshots:

N = 1000

```
public class HWQUPC_Solution {  
  
    public static int count(int n){  
  
        UF h = new UF_HWQUPC(n);  
        int count = 0;  
        Random rand = new Random();  
  
        while (h.components() > 1) {  
            count++;  
            int p = rand.nextInt(n);  
            int q = rand.nextInt(n);  
            if (p != q && !h.connected(p, q))  
                h.union(p, q);  
        }  
        return count;  
    }  
}
```

Run: HWQUPC\_Solution

2.Total number of pairs generated for 1000 objects are: 3936  
3.Total number of pairs generated for 1000 objects are: 4634  
4.Total number of pairs generated for 1000 objects are: 3427  
5.Total number of pairs generated for 1000 objects are: 3126

Average Value of number of pairs for n = 1000 is: 3685

Process finished with exit code 0

N = 2000

```
public class HWQUPC_Solution {  
  
    public static int count(int n){  
  
        UF h = new UF_HWQUPC(n);  
        int count = 0;  
        Random rand = new Random();  
  
        while (h.components() > 1) {  
            count++;  
            int p = rand.nextInt(n);  
            int q = rand.nextInt(n);  
            if (p != q && !h.connected(p, q))  
                h.union(p, q);  
        }  
        return count;  
    }  
}
```

Run: HWQUPC\_Solution

"C:\Program Files\Java\jdk-16.0.1\bin\java.exe" ...  
Running 5 times for n = 2000

1.Total number of pairs generated for 2000 objects are: 8841  
2.Total number of pairs generated for 2000 objects are: 7933  
3.Total number of pairs generated for 2000 objects are: 9454  
4.Total number of pairs generated for 2000 objects are: 10812  
5.Total number of pairs generated for 2000 objects are: 7708

Average Value of number of pairs for n = 2000 is: 8949

N = 3000

```
public class HWQUPC_Solution {  
  
    public static int count(int n){  
  
        UF h = new UF_HWQUPC(n);  
        int count = 0;  
        Random rand = new Random();  
  
        while (h.components() > 1) {  
            count++;  
            int p = rand.nextInt(n);  
            int q = rand.nextInt(n);  
            while (p == q) {  
                q = rand.nextInt(n);  
            }  
            h.union(p, q);  
        }  
        return count;  
    }  
}
```

Run: HWQUPC\_Solution

"C:\Program Files\Java\jdk-16.0.1\bin\java.exe" ...  
Running 5 times for n = 3000  
1.Total number of pairs generated for 3000 objects are: 12836  
2.Total number of pairs generated for 3000 objects are: 13651  
3.Total number of pairs generated for 3000 objects are: 11387  
4.Total number of pairs generated for 3000 objects are: 13788  
5.Total number of pairs generated for 3000 objects are: 10800  
  
Average Value of number of pairs for n = 3000 is: 12476

N = 4000

```
public class HWQUPC_Solution {  
  
    public static int count(int n){  
  
        UF h = new UF_HWQUPC(n);  
        int count = 0;  
        Random rand = new Random();  
  
        while (h.components() > 1) {  
            count++;  
            int p = rand.nextInt(n);  
            int q = rand.nextInt(n);  
            while (p == q) {  
                q = rand.nextInt(n);  
            }  
            h.union(p, q);  
        }  
        return count;  
    }  
}
```

Run: HWQUPC\_Solution

2.Total number of pairs generated for 4000 objects are: 17226  
3.Total number of pairs generated for 4000 objects are: 18565  
4.Total number of pairs generated for 4000 objects are: 19814  
5.Total number of pairs generated for 4000 objects are: 19160  
  
Average Value of number of pairs for n = 4000 is: 17875  
  
Process finished with exit code 0

N = 5000

The screenshot shows an IDE window titled "INFO6205-Assignments - HWQUPC\_Solution.java [INFO6205]". The code editor displays the following Java code:

```
public class HWQUPC_Solution {  
  
    public static int count(int n){  
  
        UF h = new UF_HWQUPC(n);  
        int count = 0;  
        Random rand = new Random();  
  
        while (h.components() > 1) {  
            count++;  
            int p = rand.nextInt(n);  
            int q = rand.nextInt(n);  
            h.union(p, q);  
        }  
    }  
}
```

The Run window shows the output of the program:

```
Running 5 times for n = 5000  
1.Total number of pairs generated for 5000 objects are: 20936  
2.Total number of pairs generated for 5000 objects are: 24936  
3.Total number of pairs generated for 5000 objects are: 22976  
4.Total number of pairs generated for 5000 objects are: 19258  
5.Total number of pairs generated for 5000 objects are: 24483  
  
Average Value of number of pairs for n = 5000 is: 22517
```

## Unit Test Screenshot:

The screenshot shows an IDE window titled "INFO6205-Assignments - HWQUPC\_Test.java [INFO6205]". The code editor displays the following Java code:

```
public class HWQUPC_Test {  
  
    public static void main(String[] args) {  
  
        int n = 100;  
        if (args.length > 0) n = Integer.parseInt(args[0]);  
  
        System.out.println(String.format("Running 5 times for n = %d", n));  
        int averageValue = 0;  
        for (int i = 0; i < 5; i++) {  
            int pairs = count(n);  
            averageValue += pairs;  
            System.out.println(String.format("%d.Total number of pairs generated for %d objects are: %d ", i+1, n, pairs));  
        }  
        System.out.println(String.format("\nAverage Value of number of pairs for n = %s is: %s", n, averageValue));  
    }  
}
```

The Run window shows the results of the unit tests:

```
Tests passed: 13 of 13 tests - 47 ms  
UF_HWQUPC_Test (edu.neu.coe.info6205.union_find) 47 ms  
  testIsConnected01 15 ms  
  testIsConnected02 0 ms  
  testIsConnected03 32 ms  
  testFind0 0 ms  
  testFind1 0 ms  
  testFind2 0 ms  
  testFind3 0 ms  
  
Process finished with exit code 0
```

