

PROGRAM STRUCTURES AND ALGORITHMS

FALL 2021

Assignment - 3(WQUPC)

Shashwat Shrey -- 002128122

Tasks Implemented :

1. Implement height-weighted Quick Union with Path Compression.
2. Use WQUPC class to get multiple values of generated pairs for N components
3. Derive a relation between number of components(N), and number of Pairs(M) generated to create a cyclic graph.

1. Code Implementation for UF_HWQUPC

```
public int find(int p) {  
    validate(p);  
    int root = p;  
    // TO BE IMPLEMENTED  
    while(root != parent[root]){  
        if (this.pathCompression) {  
            doPathCompression(root);  
        }  
        root = parent[root];  
    }  
    return root;  
}
```

```
private void doPathCompression(int i) {  
    // TO BE IMPLEMENTED update parent to value of grandparent  
    parent[i] = parent[parent[i]];  
}
```

```
private void mergeComponents(int i, int j) {  
    // TO BE IMPLEMENTED make shorter root point to taller one  
    int a = parent[i];  
    int b = parent[j];  
  
    if(height[a] < height[b]){  
        parent[a] = b;  
        height[b] += height[a];  
    }else{  
        parent[b] = parent[a];  
        height[a] += height[b];  
    }  
}
```

Passed all test cases

```
1 //...
2
3 package edu.neu.coe.info6205.union_find;
4
5 import ...
6
7
8
9
10
11
12 public class UF_HWQUPC_Test {
13
14
15     @Test
16     public void testToString() {
17         Connections h = new UF_HWQUPC( n: 2);
18         assertEquals( expected: "UF_HWQUPC:\n" +
19             " count: 2\n" +
20             " path compression? true\n" +
21             " parents: [0, 1]\n" +
22             " heights: [1, 1]", h.toString());
23     }
24
25     /**
26      *
27      */
28     @Test
29     public void testIsConnected01() {
30         Connections h = new UF_HWQUPC( n: 2);
31     }
32 }
```

Run: UF_HWQUPC_Test

Tests passed: 13 of 13 tests - 24ms

Test Case	Duration
testIsConnected01	11 ms
testIsConnected02	1 ms
testIsConnected03	3 ms
testFind0	3 ms
testFind1	1 ms
testFind2	0 ms
testFind3	3 ms
testFind4	1 ms
testFind5	0 ms
Tests passed: 13	0 ms

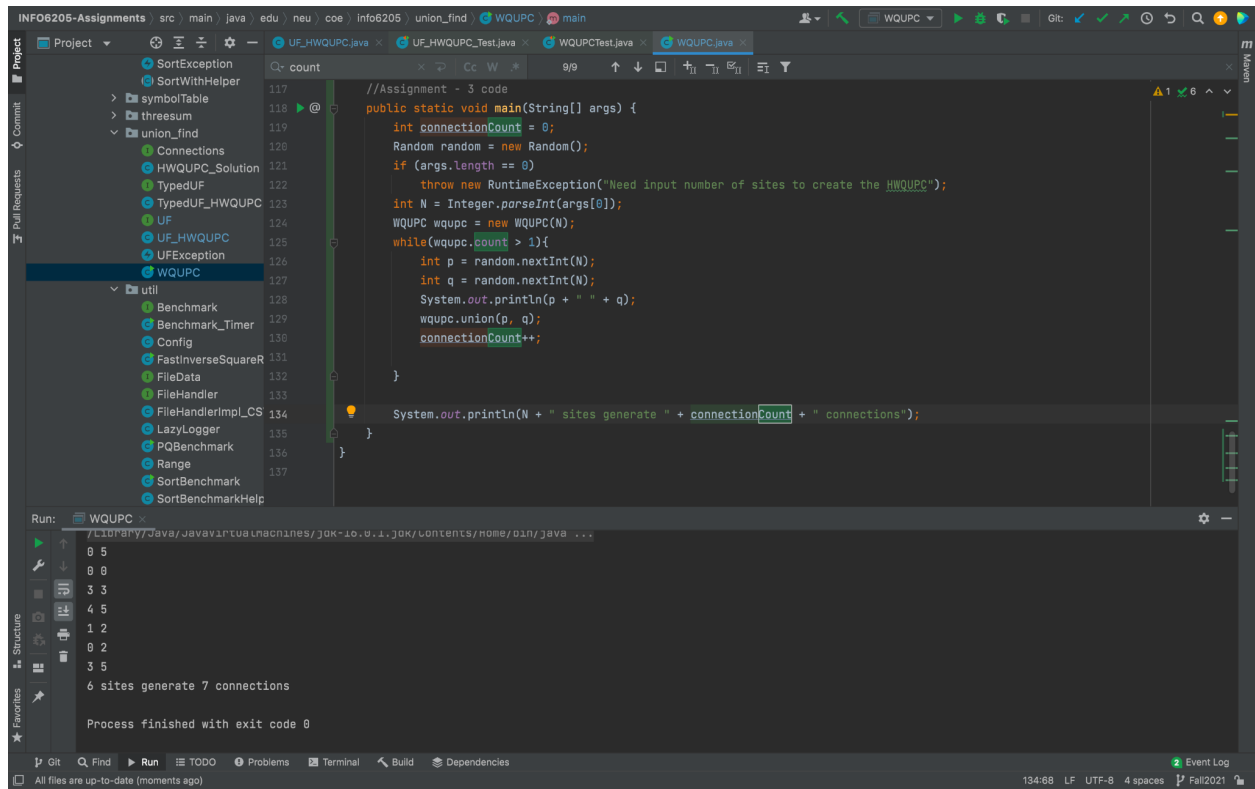
```
1 //...
2
3 package edu.neu.coe.info6205.union_find;
4
5 import ...
6
7
8
9
10
11
12 public class WQUPCTest {
13
14
15     /**
16      *
17      */
18     @Test
19     public void testFind0() {
20         WQUPC h = new WQUPC( n: 10);
21         assertEquals( expected: 0, h.find( p: 0));
22     }
23
24     /**
25      *
26      */
27     @Test
28     public void testFind1() {
29         WQUPC h = new WQUPC( n: 10);
30         assertEquals( expected: 1, h.find( p: 1));
31     }
32 }
```

Run: WQUPCTest

Tests passed: 6 of 6 tests - 9ms

Test Case	Duration
testFind0	9 ms
testFind1	0 ms
testFind2	0 ms
testFind3	0 ms
testFind4	0 ms
testConnected01	0 ms
Tests passed: 6	0 ms

2. Implemented a main method to get evidence for deriving a relation between N and M



The screenshot shows an IDE with a project named 'INFO6205-Assignments'. The left sidebar displays a file tree with a 'util' package containing various utility classes. The main editor shows the 'WQUPC.java' file, which implements a main method for generating connections between sites. The code includes a loop that generates random connections until a specified count is reached. The bottom panel shows the output of the program, which prints a series of pairs of integers representing connections between sites, followed by a summary line: '6 sites generate 7 connections'.

```
//Assignment - 3 code
public static void main(String[] args) {
    int connectionCount = 0;
    Random random = new Random();
    if (args.length == 0)
        throw new RuntimeException("Need input number of sites to create the HWQUPC");
    int N = Integer.parseInt(args[0]);
    WQUPC wqupc = new WQUPC(N);
    while(wqupc.count > 1){
        int p = random.nextInt(N);
        int q = random.nextInt(N);
        System.out.println(p + " " + q);
        wqupc.union(p, q);
        connectionCount++;
    }
    System.out.println(N + " sites generate " + connectionCount + " connections");
}
```

Run: WQUPC x
/Library/Java/JavaVirtualMachines/jdk-10.0.1.jdk/Contents/Home/bin/java ...
0 5
0 0
3 3
4 5
1 2
0 2
3 5
6 sites generate 7 connections
Process finished with exit code 0

3. Derive a Mathematical relation between N and M , $\log(\log(\text{Ratio}))$ gives 1.93 $\rightarrow N^{1.93} = M$

N	M	Ratio(M/N)	$\log(\text{Ratio})$	$\log(\log(\text{ratio}))$		
10	11	1.1	0.1375035237	-2.862459504		
20	44	2.2	1.137503524	0.1858710145		
30	94	3.133333333	1.647698256	0.7204520651		
40	187	4.675	2.224966365	1.153783527		
50	280	5.6	2.485426827	1.31349363		
60	470	7.833333333	2.969626351	1.570281417		
70	593	8.471428571	3.082605278	1.624150167		
80	812	10.15	3.343407822	1.741319343		
90	971	10.78888889	3.431474389	1.778828587		
100	1228	12.28	3.618238656	1.85528757		
110	1525	13.86363636	3.793233814	1.923428302	log(log(Ratio)) gives 1.93 $N^{1.93} = M$	
120	1807	15.05833333	3.912490195	1.968087137		
130	2133	16.40769231	4.036300437	2.013033564		
140	2692	19.22857143	4.265179678	2.092606519		
150	3166	21.10666667	4.39962685	2.137381168		
160	3563	22.26875	4.476948673	2.162515778		
170	4105	24.14705882	4.593775571	2.199680375		
180	4607	25.59444444	4.677758786	2.225817469		
190	5228	27.51578947	4.782187817	2.257670792		
200	5888	29.44	4.879705766	2.28679416		
210	6557	31.22380952	4.964574662	2.311670121		
220	7527	34.21363636	5.096499543	2.349506693		
230	8276	35.9826087	5.169227879	2.369948803		
240	9044	37.68333333	5.235854681	2.388425056		
250	9849	39.396	5.299977251	2.405986167		
260	10497	40.37307692	5.335321637	2.415575246		
270	11463	42.45555556	5.407881446	2.435063525		
280	12322	44.00714286	5.459665803	2.448812644		
290	13105	45.18965517	5.497920643	2.458886083		
300	14025	46.75	5.54689446	2.471680276		
310	14951	48.22903226	5.591829957	2.48332049		
320	15969	49.903125	5.641058257	2.495965836		
330	17345	52.56060606	5.715910006	2.514983202		
340	18576	54.63529412	5.771761321	2.529011641		
350	19569	55.91142857	5.805071302	2.537313787		
360	20832	57.86666667	5.854660637	2.549585548		
Averages =>		27.70121939	4.313604668	1.930937728		