

Program Structures and Algorithms
Spring 2023(SEC –1)
Assignment 3 : Height-weighted Quick Union with Path Compression

NAME: Shubham Sharad Bagal
NUID: 002708621

Task:

The task entails writing a Java class that implements a height-weighted Quick Union with Path compression (UF HWQUPC). This class should include methods for performing union and connected operations. The second step is to create a union-find client that generates random pairs of integers, then calls the connected() and union() methods until all sites are connected. The client should return the number of generated connections. To reduce the number of components from n to 1, the final step is to determine the relationship between the number of objects and the number of pairs generated. Based on the observations, the conclusion should be justified.

Relationship Conclusion:

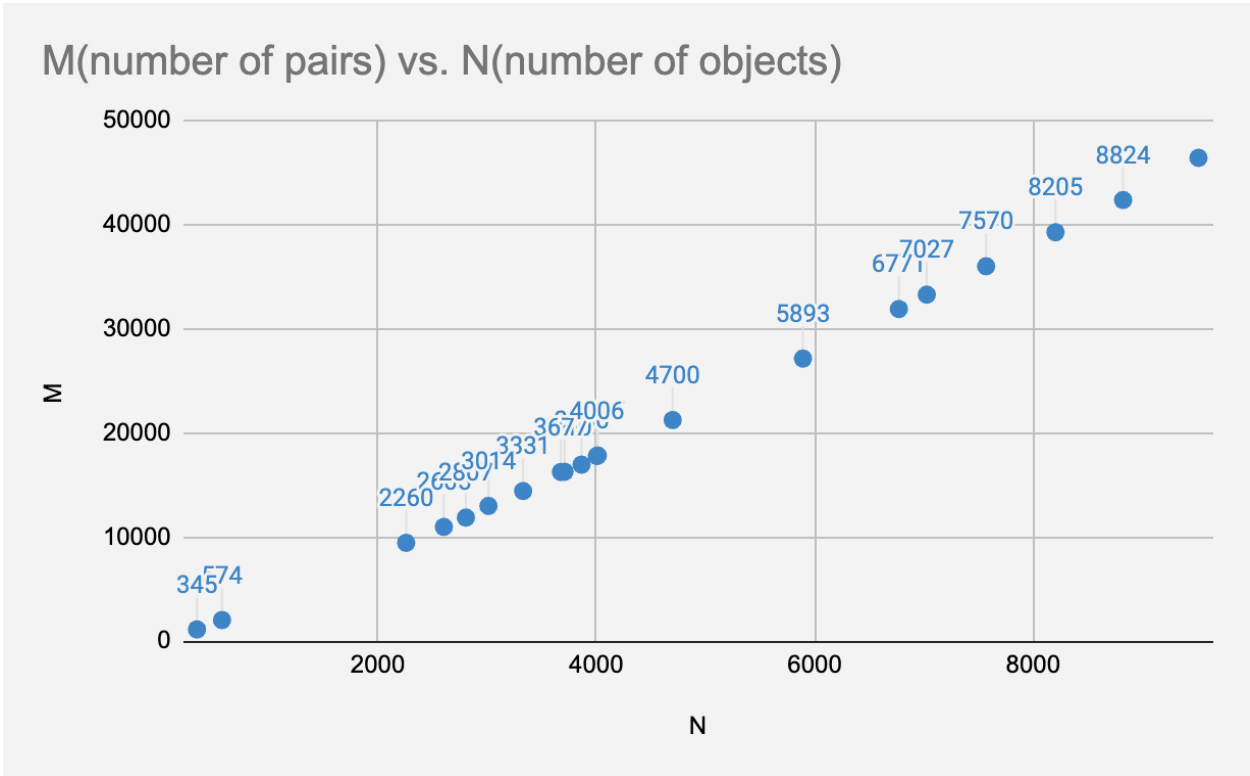
In general, the UF HWQUPC algorithm with path compression and height-weighted union has a near linear time complexity with a number of pairs proportional to $y = mx + c$.

It is important to note that this is an average-case scenario that may vary depending on the specific implementation, data structure used, and input. It does, however, provide a broad overview of the relationship between the number of nodes and the number of pairs.

Evidence to support that conclusion:

N	M
3866	16953
574	2021
6771	31910
3710	16268
4700	21240
3677	16254
4015	17830
2605	10971
345	1121
9514	46434
4006	17787
3331	14425
2260	9442
8205	39287
2807	11870
7570	36021
8824	42383
7027	33292
5893	27147
3014	12987

Graphical Representation:



Unit Test Screenshots:

The screenshot shows an IDE with the Package Explorer on the left, the HWQUPC_Solution.java file in the center, and the Console on the right. The Package Explorer shows a project named 'INFO62045_Updated [INFO62045_Updated Spring2023]' with a package 'edu.neu.coe.info6205' containing various sub-packages and classes. The HWQUPC_Solution.java file contains the following code:

```
1 package edu.neu.coe.info6205.union_find;
2
3 import java.util.Random;
4
5 public class HWQUPC_Solution {
6
7     public static int countPairs(int n) {
8         int connections = 0;
9         UF_HQUPC u = new UF_HQUPC(n);
10        Random r = new Random();
11        while (u.components() > 1) {
12            connections++;
13            int i = r.nextInt(n);
14            int j = r.nextInt(n);
15            if (!u.connected(i, j)) {
16                u.union(i, j);
17            }
18        }
19        return connections;
20    }
21
22    public static void main(String[] args) {
23        int count=0;
24        while(count<10) {
25            Random r = new Random();
26            int n = r.nextInt(10000);
27            int sum = 0;
28            for(int i = 0; i<1000; i++) {
29                sum = sum + countPairs(n);
30            }
31            System.out.println("The number of nodes (n) : " + n + " and Connections (m) : " + sum/1000);
32            count++;
33        }
34    }
35
36 }
37
38
39
40 }
```

The Console shows the output of the program, displaying the number of nodes (n) and the number of connections (m) for 10 different runs. The output is as follows:

```

The number of nodes (n) : 2861 and Connections (m) : 12133
The number of nodes (n) : 3301 and Connections (m) : 14408
The number of nodes (n) : 7434 and Connections (m) : 35318
The number of nodes (n) : 6961 and Connections (m) : 32997
The number of nodes (n) : 3344 and Connections (m) : 14471
The number of nodes (n) : 7652 and Connections (m) : 36270
The number of nodes (n) : 287 and Connections (m) : 611
The number of nodes (n) : 7331 and Connections (m) : 34866
The number of nodes (n) : 8682 and Connections (m) : 41553
The number of nodes (n) : 6803 and Connections (m) : 27721
```

The screenshot shows an IDE with the Package Explorer on the left, the HWQUPC_Solution.java file in the center, and the Console on the right. The Package Explorer shows a project named 'INFO62045_Updated [INFO62045_Updated Spring2023]' with a package 'edu.neu.coe.info6205' containing various sub-packages and classes. The HWQUPC_Solution.java file contains the following code:

```
1 package edu.neu.coe.info6205.union_find;
2
3 import java.util.Random;
4
5 public class HWQUPC_Solution {
6
7     public static int countPairs(int n) {
8         int connections = 0;
9         UF_HQUPC u = new UF_HQUPC(n);
10        Random r = new Random();
11        while (u.components() > 1) {
12            connections++;
13            int i = r.nextInt(n);
14            int j = r.nextInt(n);
15            if (!u.connected(i, j)) {
16                u.union(i, j);
17            }
18        }
19        return connections;
20    }
21
22    public static void main(String[] args) {
23        int count=0;
24        while(count<10) {
25            Random r = new Random();
26            int n = r.nextInt(10000);
27            int sum = 0;
28            for(int i = 0; i<1000; i++) {
29                sum = sum + countPairs(n);
30            }
31            System.out.println("The number of nodes (n) : " + n + " and Connections (m) : " + sum/1000);
32            count++;
33        }
34    }
35
36 }
37
38
39
40 }
```

The Console shows the output of the program, displaying the number of nodes (n) and the number of connections (m) for 10 different runs. The output is as follows:

```

The number of nodes (n) : 2861 and Connections (m) : 12133
The number of nodes (n) : 3301 and Connections (m) : 14408
The number of nodes (n) : 7434 and Connections (m) : 35318
The number of nodes (n) : 6961 and Connections (m) : 32997
The number of nodes (n) : 3344 and Connections (m) : 14471
The number of nodes (n) : 7652 and Connections (m) : 36270
The number of nodes (n) : 287 and Connections (m) : 611
The number of nodes (n) : 7331 and Connections (m) : 34866
The number of nodes (n) : 8682 and Connections (m) : 41553
The number of nodes (n) : 6803 and Connections (m) : 27721
```

