

# Program Structures and Algorithms

## Assignment No: 3

Name: Amar Nagargoje

NUID: 002273113

GitHub:[https://github.com/amarneu/INFO6205/blob/Spring2024/src/main/java/edu/neu/coe/info6205/union\\_find/UF\\_HWQUPC.java](https://github.com/amarneu/INFO6205/blob/Spring2024/src/main/java/edu/neu/coe/info6205/union_find/UF_HWQUPC.java)

### 1. Tasks:

Step 1:

(a) Implement height-weighted Quick Union with Path Compression.

Step 2:

Using your implementation of UF\_HWQUPC, develop a UF ("union-find") client that takes an integer value  $n$  from the command line to determine the number of "sites." Then generates random pairs of integers between 0 and  $n-1$ , calling `connected()` to determine if they are connected and `union()` if not. Loop until all sites are connected then print the number of connections generated. Package your program as a static method `count()` that takes  $n$  as the argument and returns the number of connections; and a `main()` that takes  $n$  from the command line, calls `count()` and prints the returned value. If you prefer, you can create a main program that doesn't require any input and runs the experiment for a fixed set of  $n$  values. Show evidence of your run(s).

Step 3:

Determine the relationship between the number of objects ( $n$ ) and the number of pairs ( $m$ ) generated to accomplish this (i.e. to reduce the number of components from  $n$  to 1). Justify your conclusion in terms of your observations and what you think might be going on.

### 2. Relationship conclusion:

The relationship between the number of objects( $n$ ) and the number of randomly generated pairs ( $m$ ) in order to reduce the number of components/objects from  $n$  to 1, in other words in order for all the pairs to be connected can be given as follows: After taking the average of the value of  $m$  over: 50 runs of the program(for the same value of  $n$ ),

we can say that  $m = c * n * \log(n)$

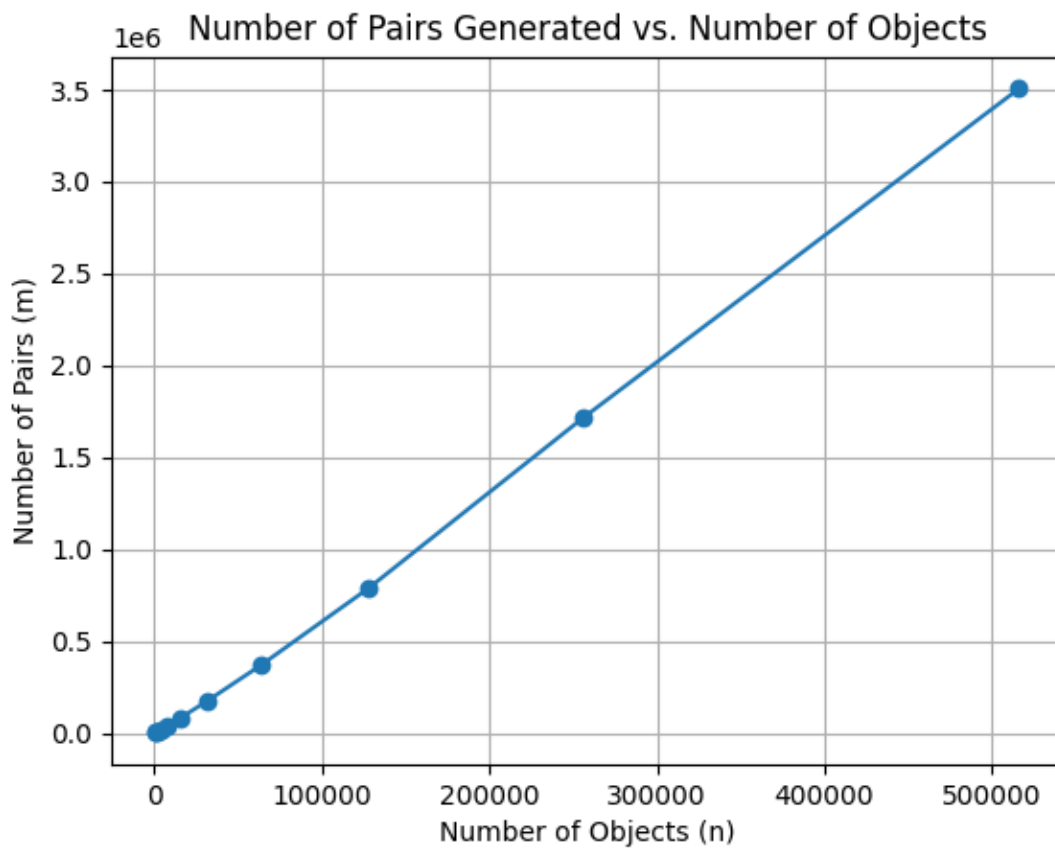
where  $c = m/n * \log(n)$

which is approximately equal to 1.22 as observed over  $n$  ranging from 500 to 516000(doubling)

Therefore, we can summarize the relationship as  $m \propto n * \log(n)$

### 3. Evidence that support conclusion:

Number of Objects (n)	No of pairs generated on avg of 50 runs	$n \log n$	$m/n \log n$
500	1677	1349.485002	1.242696286
1000	3745	3000	1.248333333
2000	7982	6602.059991	1.209016581
4000	17503	14408.23997	1.214790984
8000	37331	31224.7199	1.195559164
16000	81944	67265.91972	1.218209761
32000	174483	144164.7993	1.210302382
64000	369082	307595.5183	1.199893945
128000	786762	653722.8761	1.203509971
256000	1714288	1384509.431	1.238191638
516000	3506299	2947727.246	1.189492347



Output:

The screenshot shows an IDE with two tabs: `UF_HWQUPC.java 1, M` and `UF_HWQUPC_Test.java`. The `UF_HWQUPC.java` tab is active, displaying the following code:

```
src > main > java > edu > neu > coe > info6205 > union_find > UF_HWQUPC.java > UF_HWQUPC > main(String[])
15 public class UF_HWQUPC implements UF {
    ...
205 public static void main(String[] args){
206 {
207     int runs = 50;
208     Scanner sc = new Scanner(System.in);
209     System.out.println("Enter the number of objects: ");
210     while(sc.hasNext()) {
211         int N = sc.nextInt();
    ...
```

The `UF_HWQUPC_Test.java` tab is also visible, showing a test class with a `main` method that calls `UF_HWQUPC` with various inputs.

The terminal output shows the execution of the program, displaying the number of objects and the average number of pairs generated for 50 runs. The output is as follows:

```
amarnagargoje@Amars-MacBook-Air INFO6205 % cd /Users/amarnagargoje/Semester\ 2\6205-Program\ Structures\ and\ Algorithms\INFO6205 ; /usr/bin/env /Library/Java/JavaVirtualMachines/
jdk-11.jdk/Contents/Home/bin/java @/var/folders/cm/b30xjq2s5jdb10fm0cqmq_4c0000gn/T/cp_an30jacncr7y5ervjv8msdqtL.argfile edu.neu.coe.info6205.union_find.UF_HWQUPC
Enter the number of objects:
500
No.of objects 500 and the average of number of pairs generated for 50 runs is: 1677
Enter the number of sites:
1000
No.of objects 1000 and the average of number of pairs generated for 50 runs is: 3745
Enter the number of sites:
2000
No.of objects 2000 and the average of number of pairs generated for 50 runs is: 7982
Enter the number of sites:
4000
No.of objects 4000 and the average of number of pairs generated for 50 runs is: 17503
Enter the number of sites:
8000
No.of objects 8000 and the average of number of pairs generated for 50 runs is: 37331
Enter the number of sites:
16000
No.of objects 16000 and the average of number of pairs generated for 50 runs is: 81944
Enter the number of sites:
32000
No.of objects 32000 and the average of number of pairs generated for 50 runs is: 174483
Enter the number of sites:
64000
No.of objects 64000 and the average of number of pairs generated for 50 runs is: 369082
Enter the number of sites:
128000
No.of objects 128000 and the average of number of pairs generated for 50 runs is: 786762
Enter the number of sites:
256000
No.of objects 256000 and the average of number of pairs generated for 50 runs is: 1714288
Enter the number of sites:
512000
No.of objects 512000 and the average of number of pairs generated for 50 runs is: 3506299
Enter the number of sites:
5
```

#### 4. Unit test demonstration

← → INFO6205

UF\_HWQUPC.java M x UF\_HWQUPC\_Test.java

src > main > java > edu > neu > coe > info6205 > union\_find > UF\_HWQUPC.java > UF\_HWQUPC > doPathCompression(int)

```
15 public class UF_HWQUPC implements UF {
179 private void mergeComponents(int i, int j) {
181     if(i == j){
182         return;
183     }
184     if(height[i] < height[j]){
185         parent[i] = j;
186         height[j] += height[i];
187     }else( // height[i] >= height[j]
188         parent[j] = i;
189         height[i] += height[j];
190     )
191     // SKELETON
192     // END SOLUTION

```

PROBLEMS 419 DEBUG CONSOLE OUTPUT TERMINAL TEST RESULTS PORTS

```
%TESTE 3, testIsConnected02(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 4, testIsConnected03(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)

%TESTE 4, testIsConnected03(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 5, testFind0(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTE 5, testFind0(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 6, testFind1(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTE 6, testFind1(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 7, testFind2(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTE 7, testFind2(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 8, testFind3(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)

%TESTE 8, testFind3(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 9, testFind4(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTE 9, testFind4(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 10, testFind5(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)

%TESTE 10, testFind5(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 11, testToString(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)

%TESTE 11, testToString(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 12, testConnect01(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)

%TESTE 12, testConnect01(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 13, testConnect02(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTE 13, testConnect02(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTS 14, testConnected01(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)
%TESTE 14, testConnected01(edu.neu.coe.info6205.union_find.UF_HWQUPC_Test)

%RUNTIME74

```

Test run at 2/12/2024, 9:40:58 PM

- testConnect01()
- testConnect02()
- testConnected01()
- testFind0()
- testFind1()
- testFind2()
- testFind3()
- testFind4()
- testFind5()
- testIsConnected01()
- testIsConnected02()
- testIsConnected03()
- testToString()

Spring2024\* 0.11 Java: Warning Ln 203, Col 6 Spaces: 4 UTF-8 LF () Java Go Live Prettier