

Program Structures and Algorithms

Spring 2023 (SEC – 8)

NAME: Rishi Desai

NUID: 002751030

Assignment 4

Task:

1. (a) Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF_HWQUPC. All you have to do is to fill in the sections marked with `// TO BE IMPLEMENTED ... // ...END IMPLEMENTATION`. (b) Check that the unit tests for this class all work. You must show "green" test results in your submission (screenshot is OK).
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).
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.

Link to report:

<https://github.com/RishiDesai17/INFO6205/blob/Spring2023/assignments/assignment-4/Assignment%204.pdf>

Link to code:

https://github.com/RishiDesai17/INFO6205/tree/Spring2023/src/main/java/edu/neu/coe/info6205/union_find

Relationship Conclusion:

The number of connections will be $n - 1$. Here n is the number of sites.

The number of pairs generated (m) will be related to the number of sites (n) by the following relation:

$$m \approx \frac{n}{2} \log_e n$$

This relationship is shown by the tests performed as shown in the next section. This relationship also makes sense since it is similar to performing a sort. We check whether two points are connected or not in this case, compared to sort where we check the magnitude of two numbers.

Evidence to support that conclusion:

We start with the number of sites as 1000. This number is doubled until 512000. For each site, we perform the required operations and calculate the number of pairs generated and number of connections. On each site, these 2 metrics are calculated 20 times, and their average is taken.

```
Number of sites: 1000
Average number of connections: 999
Average number of pairs generated: 3841

Number of sites: 2000
Average number of connections: 1999
Average number of pairs generated: 8324

Number of sites: 4000
Average number of connections: 3999
Average number of pairs generated: 18455

Number of sites: 8000
Average number of connections: 7999
Average number of pairs generated: 35691

Number of sites: 16000
Average number of connections: 15999
Average number of pairs generated: 85572

Number of sites: 32000
Average number of connections: 31999
Average number of pairs generated: 173842

Number of sites: 64000
Average number of connections: 63999
Average number of pairs generated: 392737
```

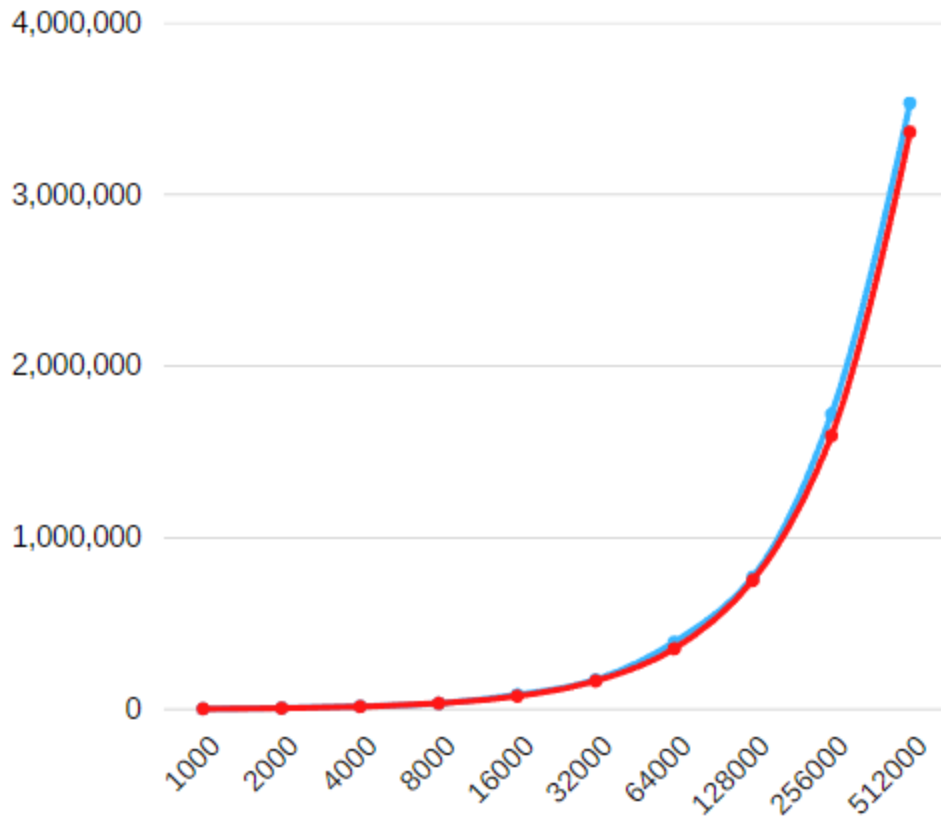
Number of sites: 128000
Average number of connections: 127999
Average number of pairs generated: 771524

Number of sites: 256000
Average number of connections: 255999
Average number of pairs generated: 1720798

Number of sites: 512000
Average number of connections: 511999
Average number of pairs generated: 3533532

n	m	$\frac{n}{2} \log_e n$
1000	3841	3453.878
2000	8324	7600.902
4000	18455	16588.1
8000	35691	35948.79
16000	85572	77442.75
32000	173842	165975.9
64000	392737	354132.4
128000	771524	752626.3
256000	1720798	1593975
512000	3533532	3365396

Graphical Representation:



The red line in the above graph is the curve for number of sites (n) vs $\frac{n}{2} \log_e n$

The blue line in the above graph is the curve for number of sites (n) vs number of pairs generated (m).

As we can see both these curves are very close to each other.

Unit Test Screenshots:

Runs: 13/13

Errors: 0

Failures: 0

edu.neu.coe.info6205.union_find.UF_HWQUPC_Test [R

testIsConnected01 (0.007 s)

testIsConnected02 (0.000 s)

testIsConnected03 (0.000 s)

testFind0 (0.000 s)

testFind1 (0.000 s)

testFind2 (0.000 s)

testFind3 (0.000 s)

testFind4 (0.000 s)

testFind5 (0.000 s)

testToString (0.000 s)

testConnect01 (0.000 s)

testConnect02 (0.000 s)

testConnected01 (0.000 s)