

Program Structures & Algorithms

Spring 2022

Assignment – 3

Name – Jayesh Kumar Khattar

NUID – 001568947

Task –

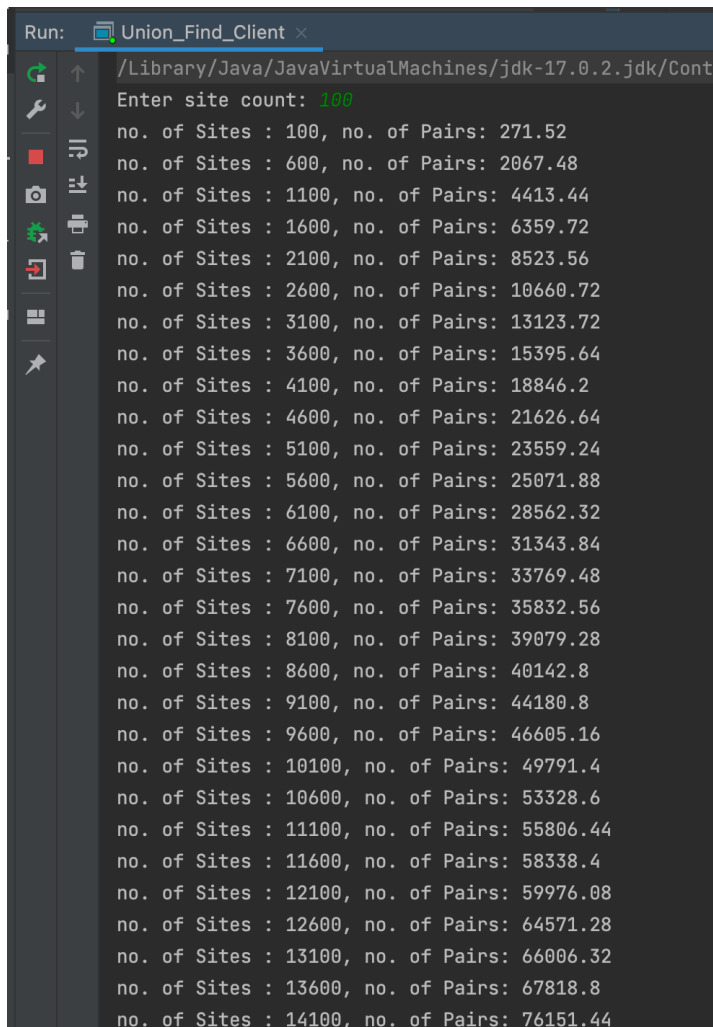
- Step 1:
 - (a) Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class `UF_HWQUPC`. All you have to do is to fill in the sections marked with `// TO BE IMPLEMENTED ... // ...END IMPLEMENTATION`.
 - (b) Check that the unit tests for this class all work. You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

Output –

Output of the above code is shown in the below picture. We have n for site count and m for no. of connections generated against each n value. We have shown value of m against different values of n below.



```
Run: Union_Find_Client x
/Library/Java/JavaVirtualMachines/jdk-17.0.2.jdk/Cont
Enter site count: 100
no. of Sites : 100, no. of Pairs: 271.52
no. of Sites : 600, no. of Pairs: 2067.48
no. of Sites : 1100, no. of Pairs: 4413.44
no. of Sites : 1600, no. of Pairs: 6359.72
no. of Sites : 2100, no. of Pairs: 8523.56
no. of Sites : 2600, no. of Pairs: 10660.72
no. of Sites : 3100, no. of Pairs: 13123.72
no. of Sites : 3600, no. of Pairs: 15395.64
no. of Sites : 4100, no. of Pairs: 18846.2
no. of Sites : 4600, no. of Pairs: 21626.64
no. of Sites : 5100, no. of Pairs: 23559.24
no. of Sites : 5600, no. of Pairs: 25071.88
no. of Sites : 6100, no. of Pairs: 28562.32
no. of Sites : 6600, no. of Pairs: 31343.84
no. of Sites : 7100, no. of Pairs: 33769.48
no. of Sites : 7600, no. of Pairs: 35832.56
no. of Sites : 8100, no. of Pairs: 39079.28
no. of Sites : 8600, no. of Pairs: 40142.8
no. of Sites : 9100, no. of Pairs: 44180.8
no. of Sites : 9600, no. of Pairs: 46605.16
no. of Sites : 10100, no. of Pairs: 49791.4
no. of Sites : 10600, no. of Pairs: 53328.6
no. of Sites : 11100, no. of Pairs: 55806.44
no. of Sites : 11600, no. of Pairs: 58338.4
no. of Sites : 12100, no. of Pairs: 59976.08
no. of Sites : 12600, no. of Pairs: 64571.28
no. of Sites : 13100, no. of Pairs: 66006.32
no. of Sites : 13600, no. of Pairs: 67818.8
no. of Sites : 14100, no. of Pairs: 76151.44
```

Relationship Conclusion –

From the above, we can see that the relationship between the number of objects (n) and the number of pairs (m) generated to reduce the number of components from n to 1 is:

$$m = \frac{1}{2} \times n \times \ln(n) 2$$

m = number of pairs generated to reduce the number of connections

n = number of sites

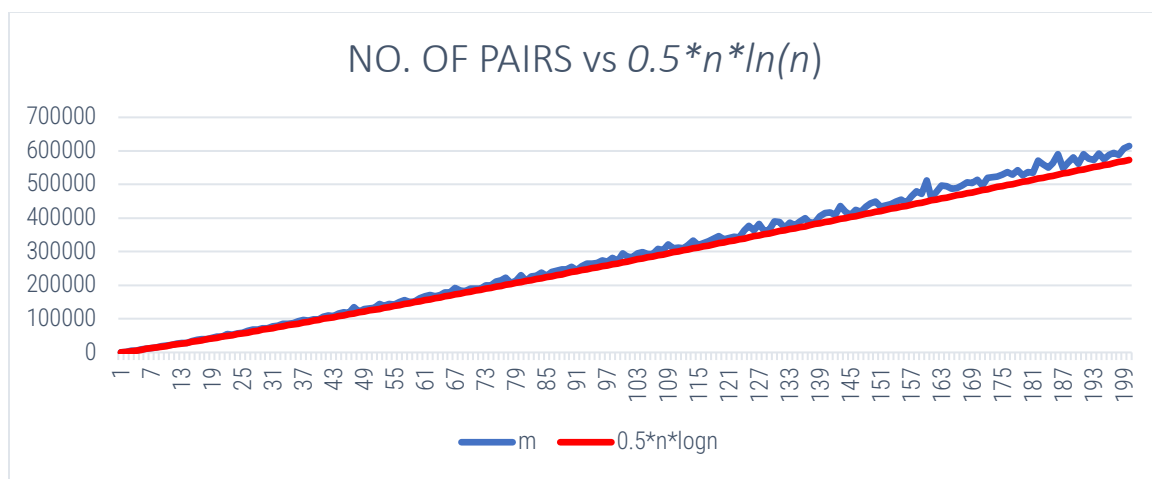
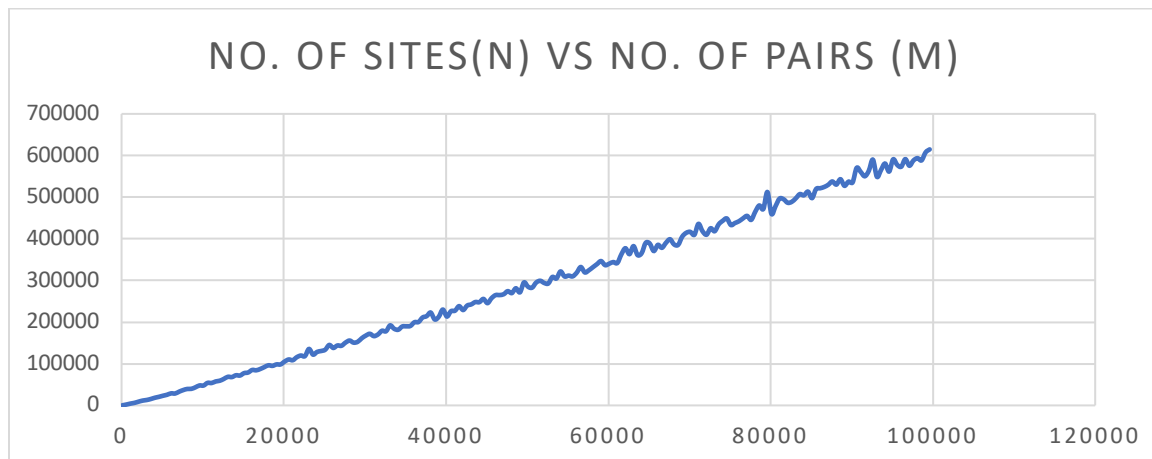
Evidence –

Let $f(N)$ be the number of pairs (m) generated to reduce the number of components from n to 1.

We took initial value of n as 100 and using the doubling method, we ran and calculated the number of pairs (m) generated to reduce the number of connections from n to 1 and computed the average number of pairs generated to accomplish this for each value of n .

In a union-find operation, we check if the pairs are connected or disconnected ($n \ln(n)$). There are only two possibilities for each pair. Hence, the relationship between m and n is almost identical $\frac{1}{2} \times n \times \ln(n)$.

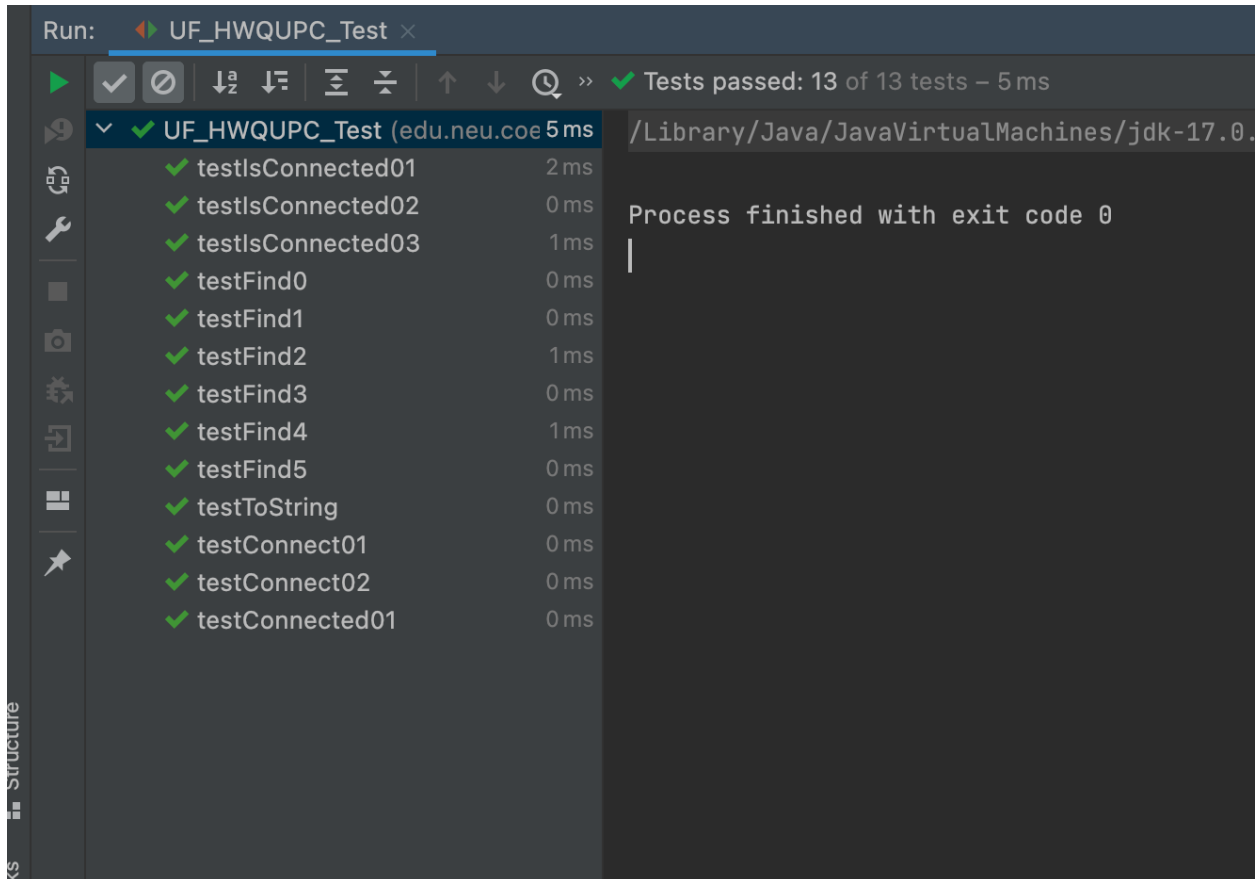
The below graphs plotted between no. of sites and no. of pairs & pair and $\frac{1}{2} \times n \times \ln(n)$ confirms the conclusion.



Unit Test Result –

Here is a screenshot of unit test results. The two classes are -

UF_HWQUPC_Test



WQUPC_Test

