

Program Structures & Algorithms

Assignment No. 2 - Benchmark

Mayannk Kumaar - 001537115

Task:

- Step 1:

(a) Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF_HWQUPC. All you must 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).

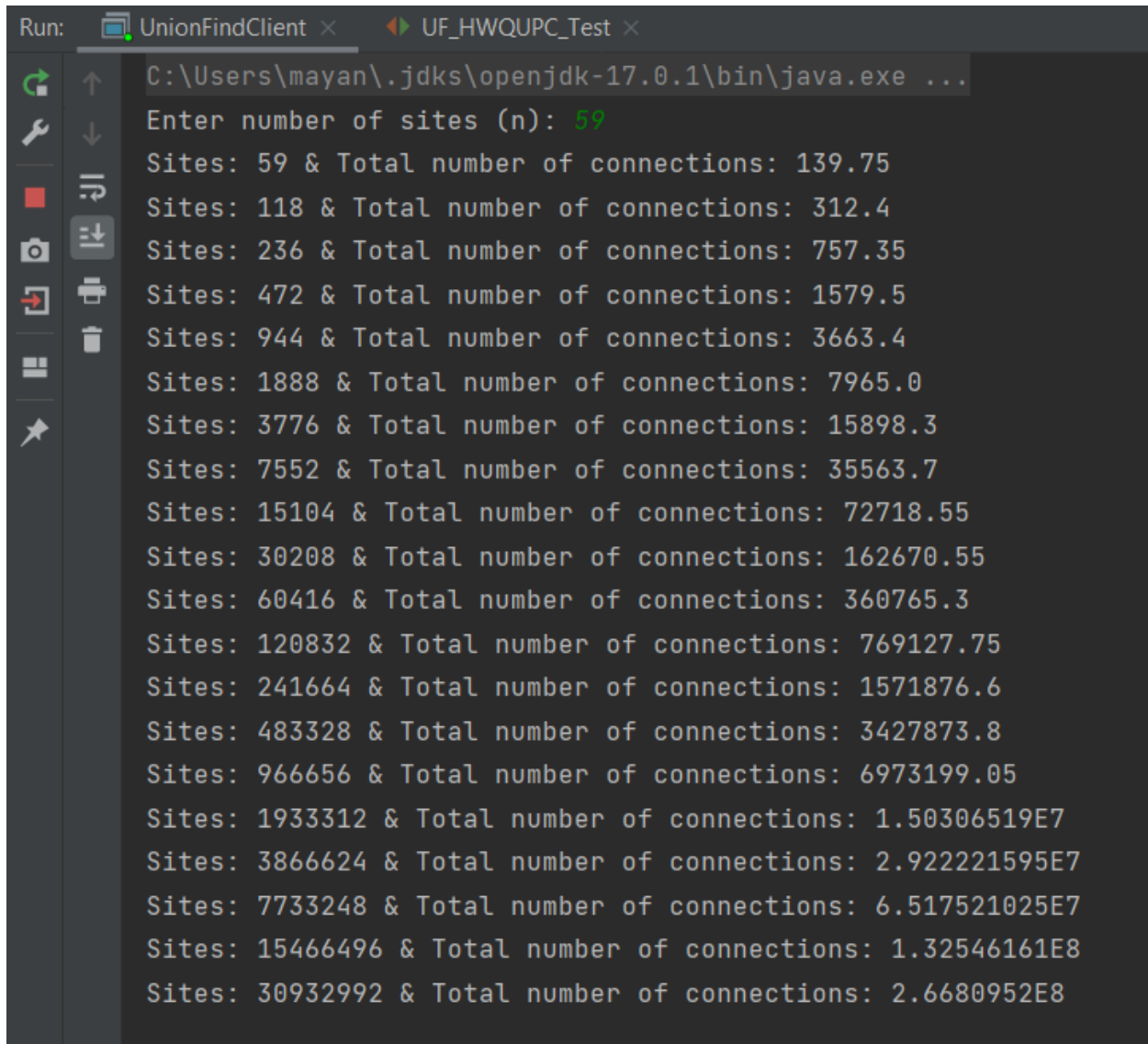
- Step 2:

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 Values:



The screenshot shows a Java application window titled "UF_HWQUPC_Test" with a dark theme. The window contains a text area with the following output:

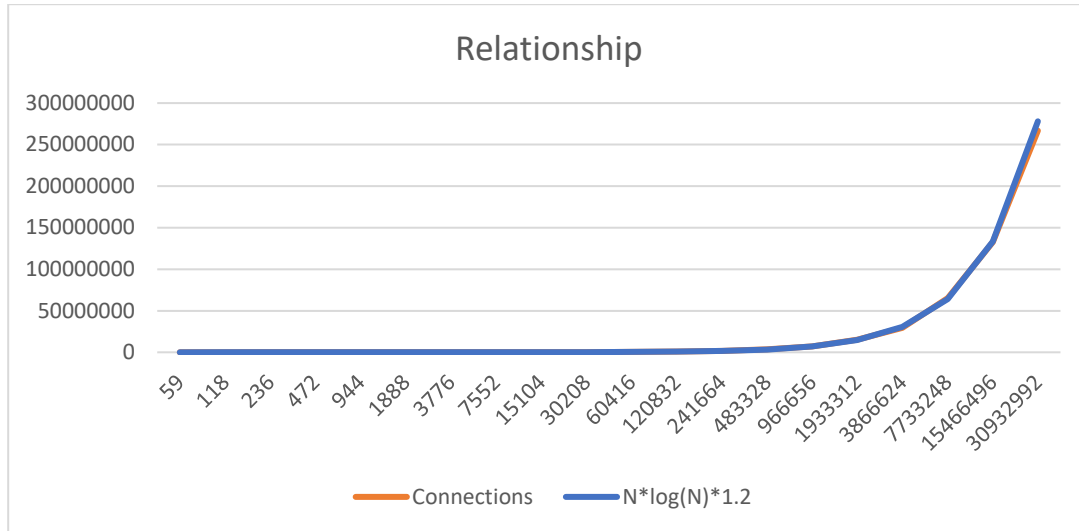
```
C:\Users\mayan\.jdk\openjdk-17.0.1\bin\java.exe ...  
Enter number of sites (n): 59  
Sites: 59 & Total number of connections: 139.75  
Sites: 118 & Total number of connections: 312.4  
Sites: 236 & Total number of connections: 757.35  
Sites: 472 & Total number of connections: 1579.5  
Sites: 944 & Total number of connections: 3663.4  
Sites: 1888 & Total number of connections: 7965.0  
Sites: 3776 & Total number of connections: 15898.3  
Sites: 7552 & Total number of connections: 35563.7  
Sites: 15104 & Total number of connections: 72718.55  
Sites: 30208 & Total number of connections: 162670.55  
Sites: 60416 & Total number of connections: 360765.3  
Sites: 120832 & Total number of connections: 769127.75  
Sites: 241664 & Total number of connections: 1571876.6  
Sites: 483328 & Total number of connections: 3427873.8  
Sites: 966656 & Total number of connections: 6973199.05  
Sites: 1933312 & Total number of connections: 1.50306519E7  
Sites: 3866624 & Total number of connections: 2.922221595E7  
Sites: 7733248 & Total number of connections: 6.517521025E7  
Sites: 15466496 & Total number of connections: 1.32546161E8  
Sites: 30932992 & Total number of connections: 2.6680952E8
```

Code:

UF_HWQUPC.java, UnionFindClient.java files attached.

Observations:

Plotting all the values of N (x-axis) and Number of connections (y-axis) along with plotting graph of $N \log(N) * 1.2$, we can clearly see it follows the same trend. Hence we can conclude that:



Number of connections = $N \log(N) * 1.2$, where N = number of sites

Sites	Connections	$N * \log(N) * 1.2$
59	139.75	125.3763224
118	312.4	293.3784922
236	757.35	672.0086792
472	1579.5	1514.520748
944	3663.4	3370.048275
1888	7965	7422.110108
3776	15898.3	16208.24733
7552	35563.7	35144.5489
15104	72718.55	75745.20626
30208	162670.55	162402.6295
60416	360765.3	346629.6928
120832	769127.75	736908.2533
241664	1571876.6	1561114.242
483328	3427873.8	3296823.955
966656	6973199.05	6942838.852
1933312	1.50E+07	14584059.59
3866624	2.92E+07	30564882.94
7733248	6.52E+07	63923293.42
15466496	1.33E+08	133433641.9
30932992	2.67E+08	278041393.9

Test Cases:

