**Program Structures and Algorithms**

**Assignment-4**

**Summer-2022**

**Dimpleben Kanjibhai Patel – 002965372**

**Task-1**:

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. Check that the unit tests for this class all work. You must show "green" test results in your submission (screenshot is OK).

**Output:**

Graphical user interface, application

Description automatically generated

**Task-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).

**Output:**

For 100 sites:

Graphical user interface, text, application

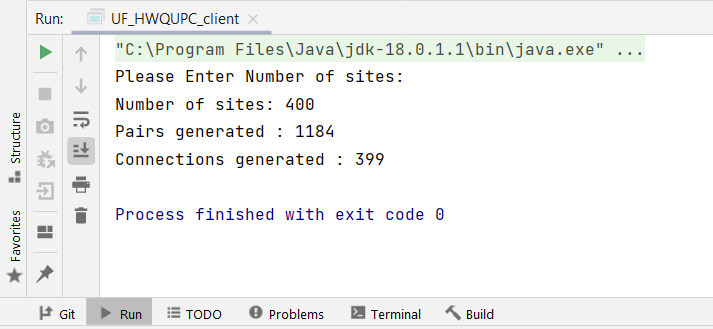
Description automatically generated

For 200 sites :

Graphical user interface, text, application, email

Description automatically generated

For 400 sites:

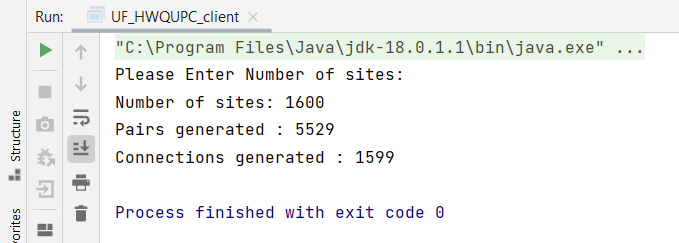


For 800 sites:

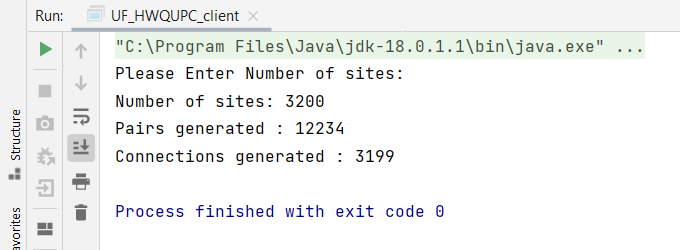
Graphical user interface, text, application

Description automatically generated

For 1600 sites:



For 3200 sites:



For 6400 sites:

Graphical user interface, text, application

Description automatically generated

For 12800 Sites:

Graphical user interface, text, application

Description automatically generated

**Task-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.

**Comparison Table:**

|  |  |  |
| --- | --- | --- |
| **N(Sties)** | **Pairs(m) Generated** | **½N(lnN)** |
| 100 | 233 | 230.25 |
| 200 | 521 | 529.38 |
| 400 | 1184 | 1198.29 |
| 800 | 2667 | 2673.84 |
| 1600 | 5529 | 5902.20 |
| 3200 | 12234 | 12913.45 |
| 6400 | 28914 | 28044.97 |
| 12800 | 62811 | 60526.08 |

**Comparison Graph:**

A picture containing line chart

Description automatically generated

**Conclusion :**

From the above simulations, it is observed that the relationship between the number of sites/objects(N) and the number of pairs (m) is,