Zixuan.Zhu (002196924)

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No. 3**

* **Task**

Step1:  
(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).

Step2:  
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).

Step3:  
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.

NOTE: although I'm not going to tell you in advance what the relationship is, I can assure you that it is a *simple* relationship.

* **Relationship Conclusion:**

The relationship between the number of objects (*n*) and the number of pairs (*m*) is close to:

*m = 0.5 \* n \* lg(n)*

* **Evidence to support the conclusion:**

1. **Output**

*表格

描述已自动生成*

Github:<https://github.com/ZixuanZhu-faye/INFO-6205/blob/8fedc32517ce8a127f5696c96582e003d4802605/INFO6205-Fall2021/src/main/java/edu/neu/coe/info6205/union_find/UF_HWQUPC.java>

For each number of sites n, “Connections” means the mean number of connections generated after 100 times running, “Expected” means we expect the connections should be 0.5 \* n \* lg(n).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Times | Sites | Connections | Expected | Ratio |
| 100 | 100 | 266.97 | 230.2585 | 1.16 |
| 100 | 300 | 984.6 | 855.5674 | 1.15 |
| 100 | 900 | 3328.41 | 3061.0776 | 1.07 |
| 100 | 1200 | 4502.83 | 4254.0461 | 1.06 |
| 100 | 1800 | 7296.06 | 6745.9877 | 1.08 |
| 100 | 2800 | 11681.68 | 11112.3246 | 1.05 |
| 100 | 3700 | 16068.98 | 15199.7630 | 1.06 |
| 100 | 5600 | 25810.81 | 24165.4613 | 1.07 |
| 100 | 7800 | 37905.11 | 34951.3281 | 1.08 |
| 100 | 9000 | 44370.56 | 40972.4094 | 1.08 |

According to the output from the program, the conclusion on the relationship between the number of objects (*n*) and the number of pairs (*m*) has been proved.

1. **Graphical Representation**

* **Unit tests result:**

图形用户界面, 应用程序, Excel

描述已自动生成