Shibo Lu (001548075)

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No. 3**

**Part 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).

**Part 1 Unit tests:**

UF\_HWQUPC\_Test：

图形用户界面, 应用程序

描述已自动生成

**Part 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).

**Part 2:**

**method count():**

文本, 信件

描述已自动生成

**method main():**

图形用户界面, 文本, 应用程序

描述已自动生成

**Git：**

https://github.com/ShiboLu/INFO6205-Shibo-Lu/blob/main/INFO6205-Fall2021/src/main/java/edu/neu/coe/info6205/union\_find/UF\_HWQUPC.java

**Output:**

图形用户界面, 应用程序

描述已自动生成

**Part 3:**

Implement a main program (or you could do it via your own unit tests) to actually run the following benchmarks: measure the running times of this sort, using four different initial array ordering situations: random, ordered, partially-ordered and reverse-ordered. I suggest that your arrays to be sorted are of type Integer. Use the doubling method for choosing n and test for at least five values of n. Draw any conclusions from your observations regarding the order of growth.

**Relationship Conclusion:**

**Evidence to support the conclusion:**

Every times we connect two 'site', It will be more difficult to find next unconnected pairs. This will make the average connect times to for each pair. After analysis with the graph of result, the relationship will be close to . After using excel to compute the average for each generated ‘point’. The coefficients of the relationship equation is 0.37114.

**Graphical Representation:**

图表, 折线图

描述已自动生成

表格, Excel

描述已自动生成

**Output:**

图形用户界面, 应用程序

描述已自动生成

**Git:**

https://github.com/ShiboLu/INFO6205-Shibo-Lu/tree/main/assignment3