**Ripan Halder**

**NU ID: 001442993**

**Assignment 3 - Height-weighted Quick Union with Path Compression.**

1. **Code ->**

<https://github.com/RipanHalder/INFO6205>

*Asignment3 Part 1 & 2 -> /src/main/java/edu/neu/coe/info6205/union\_find/UF\_HWQUPC.java*

*Results -> results/union\_find/assignment3.csv*

1. **Screenshots and Evidences:**

I ran the code for various values of n ranging from 10 – 640. The pairs generated are mean of 1000 successful runs.

The following table shows 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).

|  |  |  |
| --- | --- | --- |
| **number of objects (n)** | **number of pairs (m)** | **Formula** |
| 10 | 16 | 13.28771238 |
| 20 | 37 | 34.57542476 |
| 40 | 89 | 85.15084952 |
| 80 | 201 | 202.301699 |
| 160 | 452 | 468.6033981 |
| 320 | 1017 | 1065.206796 |
| 640 | 2247 | 2386.413592 |
|  |  |  |
|  |  |  |
|  |  |  |
| Formula = α \* N \* log2N  (where α ~ 0.4) | |  |
|  |

At first, I derived to formula by checking the trend which was logarithmic for ranging values of n. The above results of pairs were derived for every n for 1000 runs each. The final M pairs are an average of these 1000 runs. After getting the logarithmic observation it was necessary to find the factor multiplying logN. There was a constant co-efficient α multiplied with Nlog2N. After further analysis, I deduced it to be around α = ~ 0.4. For better understanding see the below chart:

Chart, line chart

Description automatically generated

Unit Test Cases - Successful Run:

Graphical user interface, text, application, table

Description automatically generated