Program Structures and Algorithms

Spring 2024

NAME: Kartikey Vijayakumar Hebbar

NUID: 002276938

GITHUB LINK: https://github.com/kartikeyhebbar/INFO6205/tree/Spring2024/src/main/java/edu/neu/coe/info6205/union\_find

**Task: Assignment 4 (WQUPC)**

**Relationship Conclusion:**

We are considering two parameters here: No. of sites (N) and Number of connections generated (M). Our goal is to check how many connections are generated by the height-weighted Quick Union Path Compression algorithm when the sites or number of nodes are changed.

In worst case, the height of the tree is logN (log base 2) and if N union operations are performed, we’ll end up with a relation where:

M ∝ N \* logN

We can re-write this equation with the help of a constant as:

M = c \* N \* logN

Where ‘c’ can be shown as:

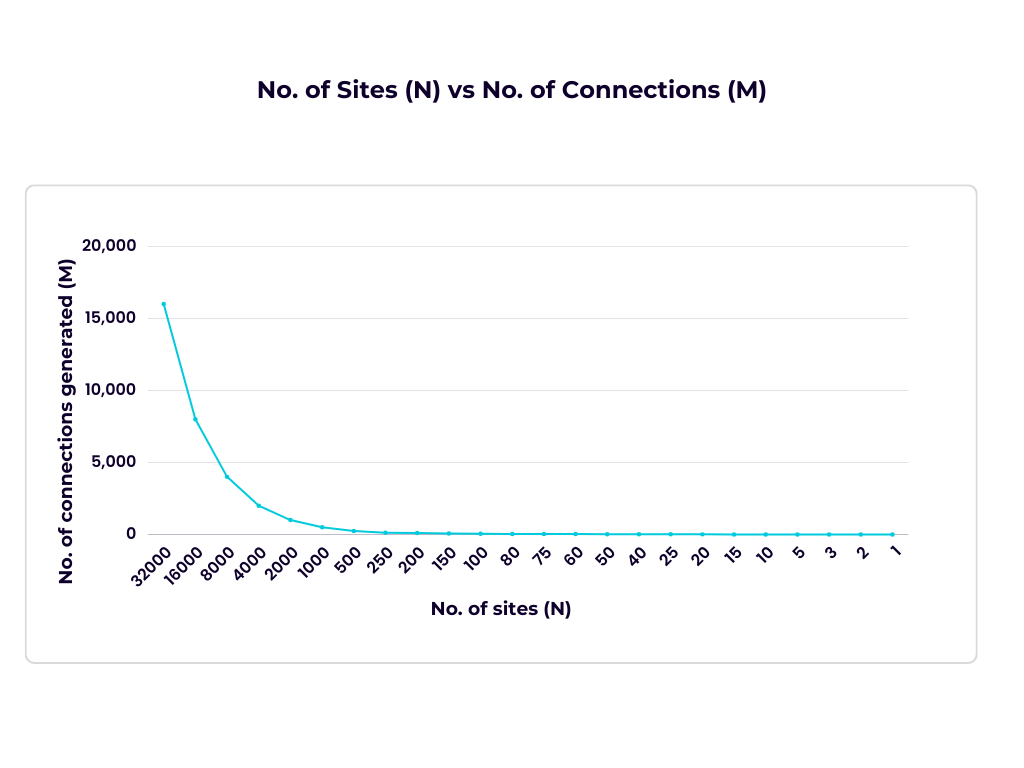
c = M / N \* logN

**Evidence to support that conclusion:**

To provide evidence for the above conclusion, we take into consideration various values of N and record the corresponding Number of Connections Generated (M). We will also compute N\*logN values for each N value and compute the constant ‘c’ value for all values of N.

|  |  |  |  |
| --- | --- | --- | --- |
| **No. of Sites (N)** | **Number of connections generated (M)** | **N log N** | **C = M / N log N** |
| 32000 | 16004 | 478692 | 0.033432771 |
| 16000 | 8001 | 224000 | 0.03571875 |
| 8000 | 4003 | 104000 | 0.038490385 |
| 4000 | 2003 | 48000 | 0.041729167 |
| 2000 | 1001 | 22000 | 0.0455 |
| 1000 | 500 | 10000 | 0.05 |
| 500 | 251 | 4500 | 0.055777778 |
| 250 | 129 | 2000 | 0.0645 |
| 200 | 101 | 1529 | 0.066056246 |
| 150 | 76 | 1083 | 0.070175439 |
| 100 | 51 | 664 | 0.076807229 |
| 80 | 41 | 506 | 0.081027668 |
| 75 | 38 | 466 | 0.081545064 |
| 60 | 31 | 354 | 0.087570621 |
| 50 | 26 | 282 | 0.092198582 |
| 40 | 21 | 213 | 0.098591549 |
| 25 | 13 | 116 | 0.112068966 |
| 20 | 11 | 86 | 0.127906977 |
| 15 | 7 | 61 | 0.114754098 |
| 10 | 5 | 33 | 0.151515152 |
| 5 | 2 | 12 | 0.166666667 |
| 3 | 2 | 5 | 0.4 |
| 2 | 1 | 2 | 0.5 |
| 1 | 0 | 0 | 0 |

Based on the above computed data, the value of the constant ‘c’ is **≈0.0938**  
The relation can be analyzed with the below graph as well:



In general, the value of M depends on N \* log N value but also depends on various other factors such as generation of random connections which can alter the performance of the algorithm.

**Unit Test Screenshots:** Screenshot of the successful unit tests:  
A screenshot of a computer

Description automatically generated

Output of the code:  
A screenshot of a computer

Description automatically generated