

David Nallapu (001530978)

## Program Structures & Algorithms

Spring 2021

### Assignment No. 4

- **Task**

Implement height-weighted Quick Union with Path Compression. Using my implementation of UF\_HWQUPC to develop a UF\_client that takes an integer values of  $n$  to determine the number of “sites.” For weighted quick union, store the depth rather than the size. For weighted quick union with path compression, do two loops, so that all intermediate nodes point to the root, not just the alternates. Coding the alternative and benchmarking it against the implementation in the repository.

- **Relationship Conclusion:**

The number of pairs ( $m$ ) is approximately  $n \cdot \log(n)/2$  where  $n$  is the number of objects. This seems to be the same even with the height and two loops modification to connect to the root.

```

<terminated> UF_Client [Java Application] /Library/Internet Plu
Weighted Quick Union(Depth)
n      m      (n*Lg(n))/2
-----
1000   3850   3453
2000   8127   7600
4000   17634  16588
8000   37825  35948
16000  79905   77442
32000  188989  165975
64000  375673  354132
128000 795127  752626
256000 1691106 1593975
512000 3557311 3365396

Weighted Quick Union(Height)
n      m      (n*Lg(n))/2
-----
1000   3453   3453
2000   7352   7600
4000   18014  16588
8000   38078  35948
16000  87162   77442
32000  178359  165975
64000  367167  354132
128000 762644  752626
256000 1643601 1593975
512000 3577296 3365396

Weighted Quick Union(intermediate nodes point
n      m      (n*Lg(n))/2
-----
1000   3521   3453
2000   7978   7600
4000   16505  16588
8000   37949  35948
16000  82612   77442
32000  178324  165975
64000  388415  354132
128000 793091  752626
256000 1698193 1593975
512000 3483922 3365396

```

- Evidence to support the conclusion(Output):
- Unit tests result(Modified with two loops, height):

