**INFO 6205**

**Program Structures & Algorithms**

**Fall 2020**

**Assignment No 4**

* **Task**
* For weighted quick union, store the depth rather than the size;
* For weighted quick union with path compression, do simpler one-pass variant (path-halving) alternates, make every other node in path point to its grandparent.

For both of these, also, benchmark them.

* **Output**
* Weighted Quick Union (store with depth):

Number of sites n = 10

union (4, 1)

union (1, 4)

union (5, 7)

union (9, 1)

union (5, 2)

union (0, 8)

union (6, 4)

union (2, 7)

union (2, 1)

union (8, 6)

union (1, 0)

union (5, 0)

union (5, 5)

union (1, 1)

union (5, 7)

union (7, 0)

union (7, 6)

union (1, 7)

union (1, 1)

union (4, 7)

union (4, 3)

WQU:

count: 1

parents: [5, 4, 5, 5, 9, 5, 9, 5, 0, 5]

depths: [1, 2, 1, 1, 2, 0, 2, 1, 2, 1]

Initiate with 10 sites and generate 21 connections

* Weighted Quick Union with Path Compression (one-pass variant):

Number of sites n = 10

union (4, 4)

union (2, 9)

union (6, 0)

union (4, 6)

union (1, 2)

union (3, 7)

union (6, 2)

union (5, 2)

union (3, 7)

union (7, 9)

union (1, 3)

union (9, 1)

union (6, 8)

WQUPC:

count: 1

parents: [6, 6, 6, 6, 6, 6, 6, 3, 6, 6]

sizes: [1, 1, 1, 2, 1, 1, 10, 1, 1, 1]

Initiate with 10 sites and generate 13 connections

* **Conclusion**
* **Evidence to support**
* **Screenshot of Unit test passing**