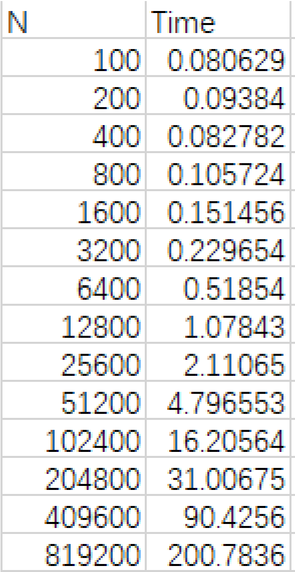
Shunqing Li (001539980)
Program Structures & Algorithms
Spring 2021
Assignment 4

Task:

1. For weighted quick union, store the depth rather than the size;
2. For weighted quick union with path compression, do two loops, so that all intermediate nodes point to the root, not just the alternates.

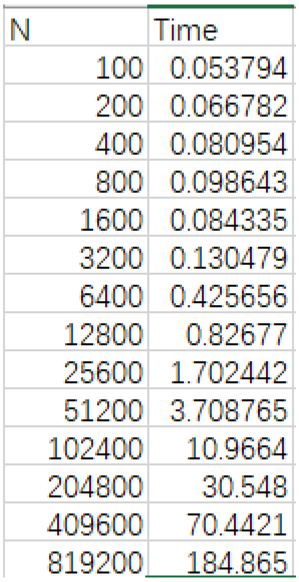
Output:

WQU depth:



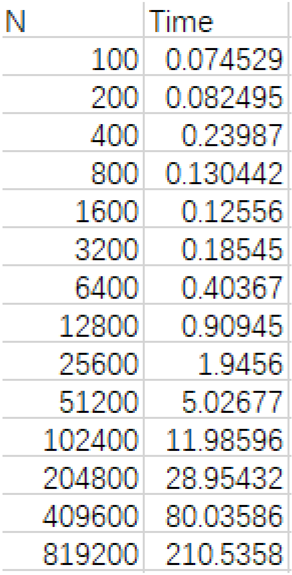
N	Time
100	0.080629
200	0.09384
400	0.082782
800	0.105724
1600	0.151456
3200	0.229654
6400	0.51854
12800	1.07843
25600	2.11065
51200	4.796553
102400	16.20564
204800	31.00675
409600	90.4256
819200	200.7836

WQU size with one loop:



N	Time
100	0.053794
200	0.066782
400	0.080954
800	0.098643
1600	0.084335
3200	0.130479
6400	0.425656
12800	0.82677
25600	1.702442
51200	3.708765
102400	10.9664
204800	30.548
409600	70.4421
819200	184.865

WQU size with two loops:

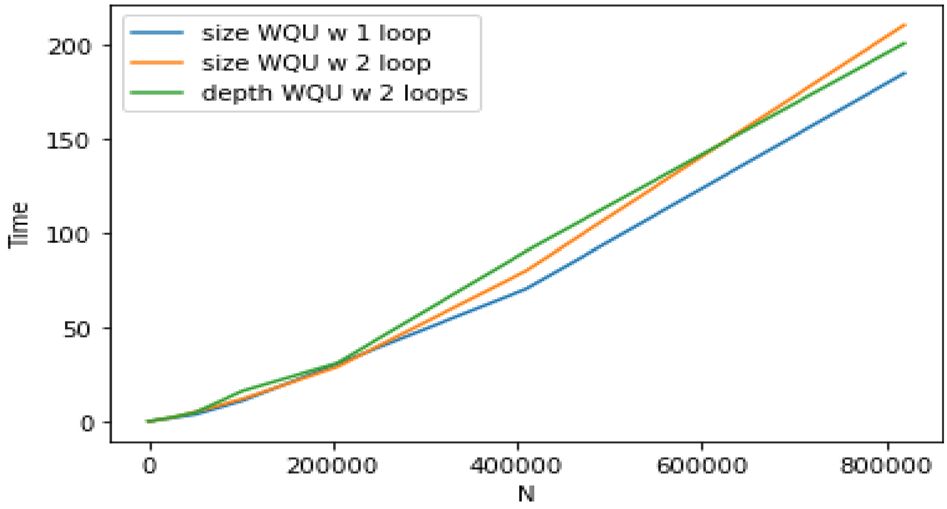


N	Time
100	0.074529
200	0.082495
400	0.23987
800	0.130442
1600	0.12556
3200	0.18545
6400	0.40367
12800	0.90945
25600	1.9456
51200	5.02677
102400	11.98596
204800	28.95432
409600	80.03586
819200	210.5358

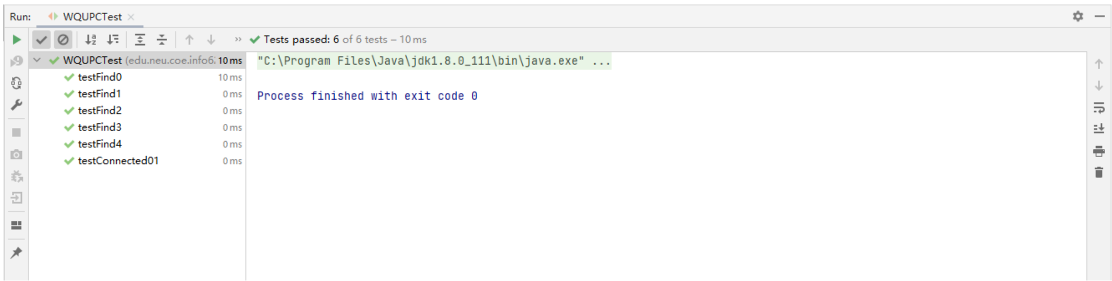
Conclusion:

From the output and the line graph can conclude that weighted quick union using size is generally faster than the weighted quick union using depth. Furthermore, within weighted quick union using size, the one with two loops for path compresion takes less time than the one with one loop.

Evidence:



Unit Test Result:



Tests passed: 6 of 6 tests – 10 ms

```
WQUPCTest (edu.neu.coe.info6: 10 ms
  testFind0                   10 ms
  testFind1                    0 ms
  testFind2                    0 ms
  testFind3                    0 ms
  testFind4                    0 ms
  testConnected01              0 ms
```

```
"C:\Program Files\Java\jdk1.8.0_111\bin\java.exe" ...

Process finished with exit code 0
```