

INFO 6205

Program Structures & Algorithms

Fall 2020

Assignment 4

Task

Code & benchmark the alternatives for union find i.e. weighted quick union with depth & weighted quick union with path compression (grandparent fix).

Evidence

#1 – Weighted Quick Union with depth

#2 – Weighted Quick Union with Path Compression (grandparent fix)

Execution of #1 & #2 code using various values of components (N) and comparing output for benchmarking runtimes.

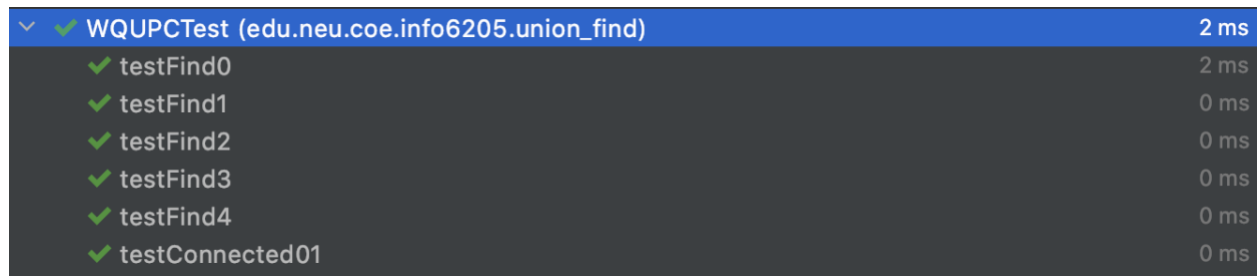
N	#1	#2
100000	30.13	24.57
200000	67.44	59.37
400000	153.14	130.84
800000	368.86	322.82

Figure 1 : Output benchmark run times for #1 & #2

Conclusion

As evident from benchmarking both the alternatives, #2 offers better runtime performance than #1 as number of objects increases. The path compression in #2 significantly improves the performance of the algorithm.

Screenshots



A screenshot of a unit test runner interface. The top row shows a collapsed test suite 'WQUPCTest (edu.neu.coe.info6205.union_find)' with a green checkmark and a duration of '2 ms'. Below it, six individual test methods are listed, each with a green checkmark and a duration of '0 ms': 'testFind0', 'testFind1', 'testFind2', 'testFind3', 'testFind4', and 'testConnected01'.

✓ WQUPCTest (edu.neu.coe.info6205.union_find)	2 ms
✓ testFind0	2 ms
✓ testFind1	0 ms
✓ testFind2	0 ms
✓ testFind3	0 ms
✓ testFind4	0 ms
✓ testConnected01	0 ms

Figure 2: Execution of Unit Tests for WUPC