Charmi Dalal (001582441)

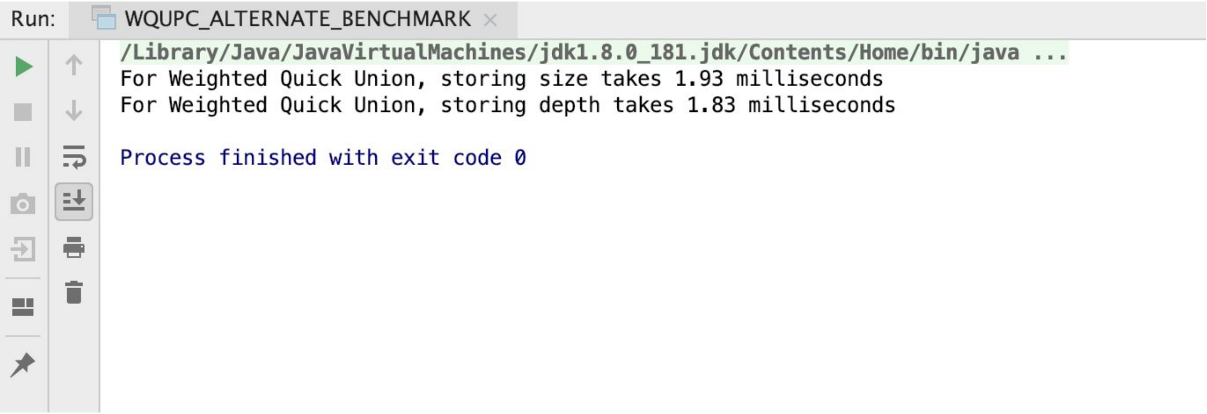
INFO 6205 Program Structure & Algorithms
Spring 2021
Assignment 4

Task 1:

For weighted quick union, store the depth rather than the size. If you can explain why alternative #1 is unnecessary to be benchmarked, you may skip benchmarking that one.

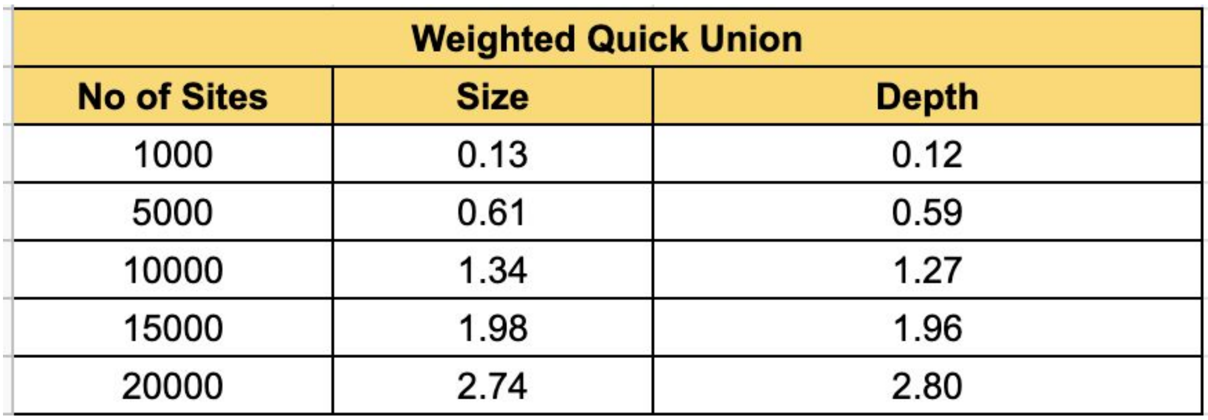
Output:

I have generated output for a different number of sites and benchmarked it. It is unnecessary to benchmark weighted unions by depth because the running time for both by size and depth are **approximately the same**. So storing by depth doesn't help in reducing run time.



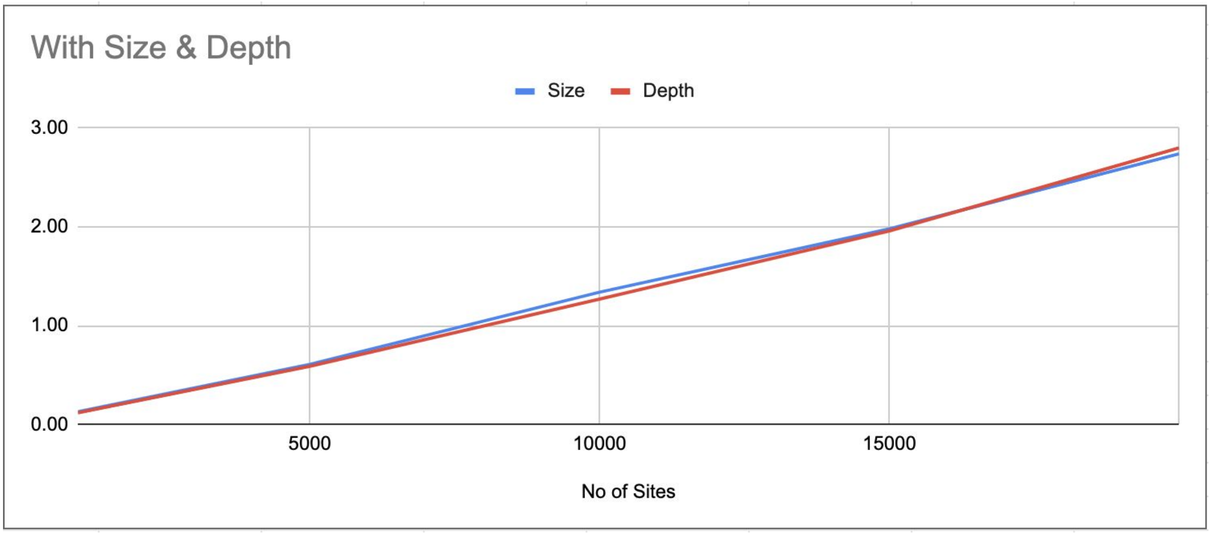
```
/Library/Java/JavaVirtualMachines/jdk1.8.0_181.jdk/Contents/Home/bin/java ...
For Weighted Quick Union, storing size takes 1.93 milliseconds
For Weighted Quick Union, storing depth takes 1.83 milliseconds

Process finished with exit code 0
```

1.1 Benchmark Results



Weighted Quick Union		
No of Sites	Size	Depth
1000	0.13	0.12
5000	0.61	0.59
10000	1.34	1.27
15000	1.98	1.96
20000	2.74	2.80

1.2 Time taken in (ms)



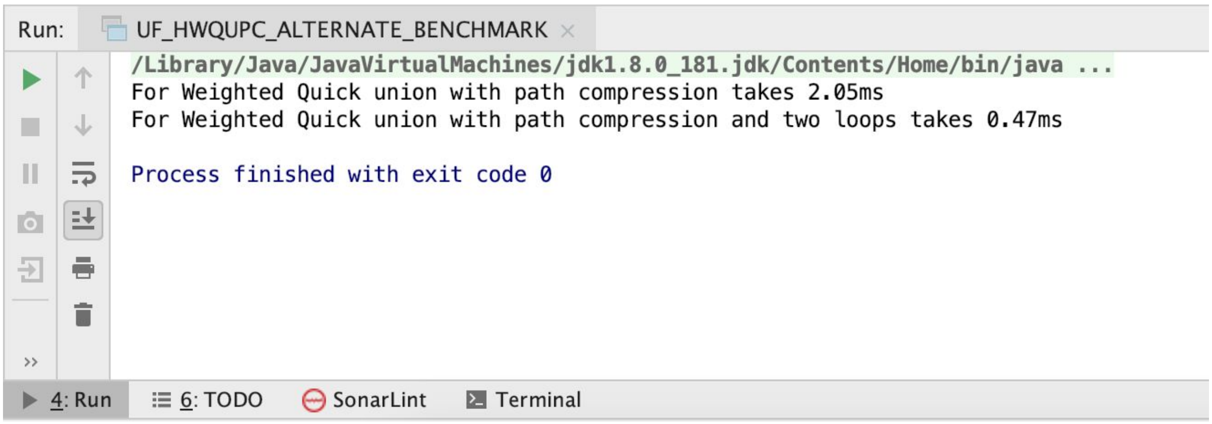
1.3 Time taken in (ms) vs No of Sites

Task 2:

For weighted quick union with path compression, do two loops, so that all intermediate nodes point to the root, not just the alternates.

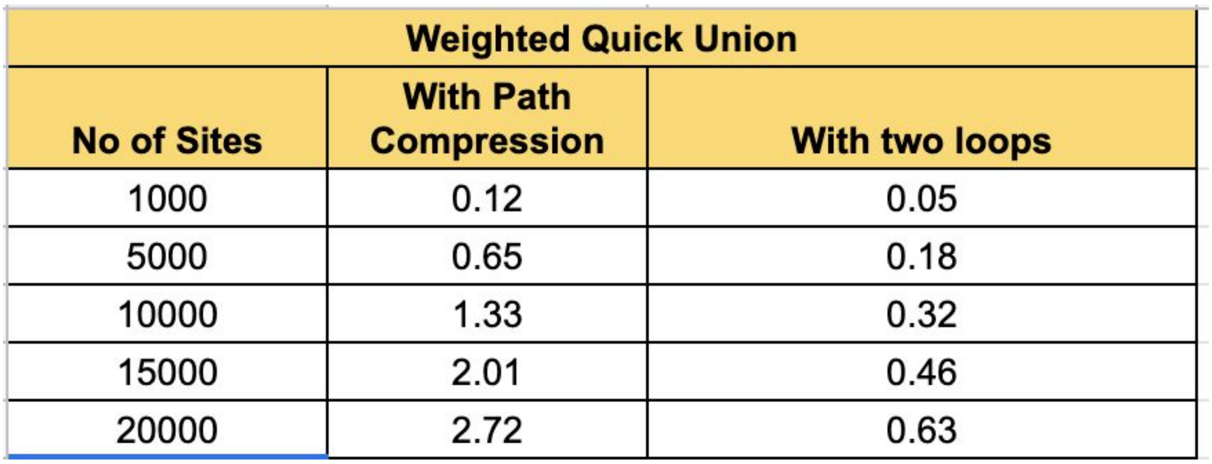
Output:

Weighted union with Path compression for different sites and with two loops improves the runtime performance compared to the previous solution.



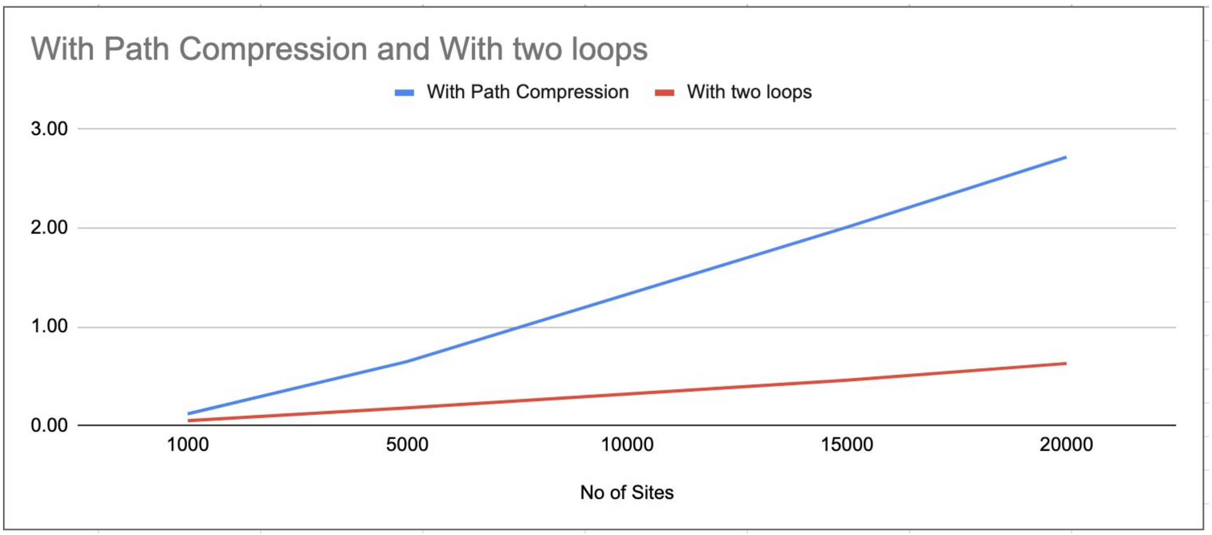
```
/Library/Java/JavaVirtualMachines/jdk1.8.0_181.jdk/Contents/Home/bin/java ...
For Weighted Quick union with path compression takes 2.05ms
For Weighted Quick union with path compression and two loops takes 0.47ms

Process finished with exit code 0
```

2.1 Benchmark Results



Weighted Quick Union		
No of Sites	With Path Compression	With two loops
1000	0.12	0.05
5000	0.65	0.18
10000	1.33	0.32
15000	2.01	0.46
20000	2.72	0.63

2.2 Time taken in (ms)



2.3 Time taken in (ms) vs Num of Sites