Sayali Mahajan – (001576540)

Program Structures & Algorithms
Summer 2021
Assignment No. 4

Task 1

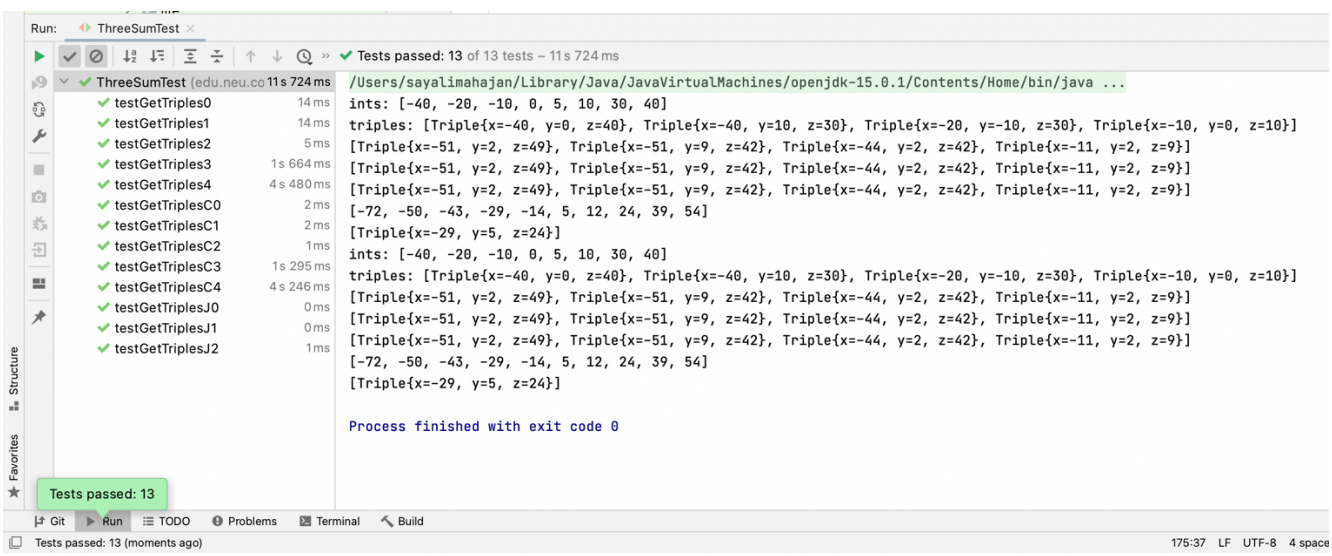
If you haven't already, code the quadratic solution of 3-SUM, i.e.

ThreeSumQuadratic

In the repository. Show that the unit tests all pass. And show a graph of your observations (use the *Benchmark* code) for at least five different (doubling) values of *N*. If the growth is not O(N^2), please explain why.

Output

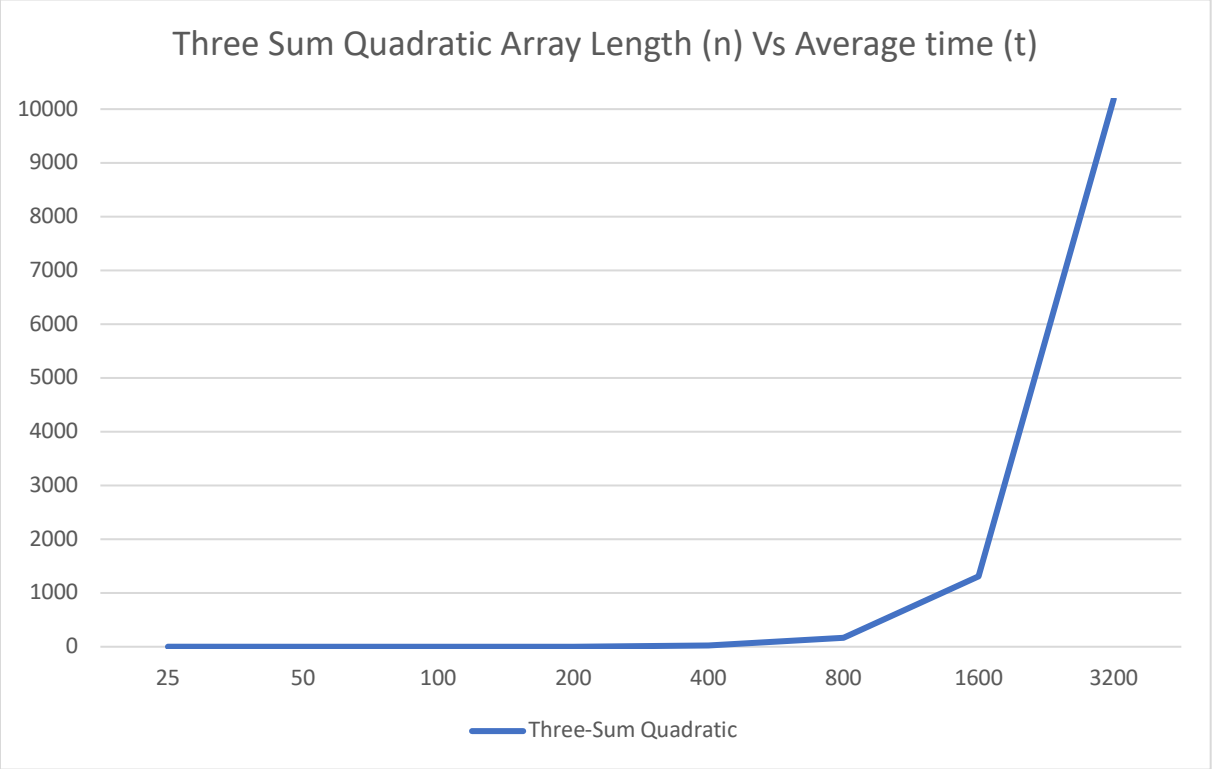
(benchmarking code for threesum problem i.e. main method is added in ThreesumQuadratic.java class)



Evidence to support conclusion

Array length	Avg time
25	0.578
50	0.602
100	0.828
200	3.272
400	23.149
800	169.872
1600	1307.194
3200	10191.822

Graphical representation



Conclusion

From multiple experiments carried out , we generated above evidence and graphical representation and based on above data analysis, It is observed that three sum quadratic solution has growth of O(N^2).

Task 2

We mentioned two alternatives for implementing Union-Find:

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.

For both of these, code the alternative and benchmark it against the implementation in the repository. You have all of that available from a previous assignment.
If you can explain why alternative #1 is unnecessary to be benchmarked, you may skip benchmarking that one.

Output

1. Storing depth rather than size

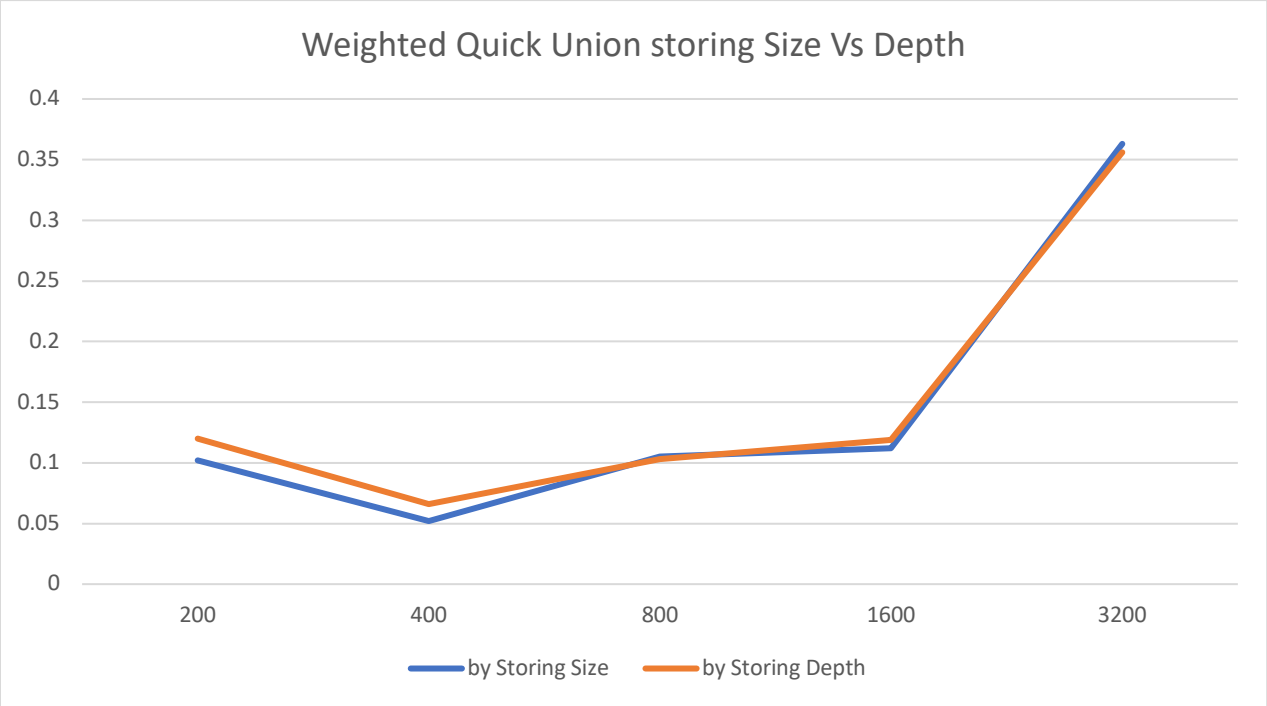
(WQUPC_alternate1 and WQUPC_alternate1_benchmark classes added for this part)

```
Run:  WQUPC_alternate1_benchmark
/Users/sayalimahajan/Library/Java/JavaVirtualMachines/openjdk-15.0.1/Contents/Home/bin/java -javaagent:/Applicat
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing size for 200 sites takes 0.10199665999999999 milliseconds.
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing depth for 200 sites takes 0.12056125 milliseconds.
-----------------------------------------------------------------------------------------------------
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing size for 400 sites takes 0.052429579999999996 milliseconds.
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing depth for 400 sites takes 0.06666042 milliseconds.
-----------------------------------------------------------------------------------------------------
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing size for 800 sites takes 0.10500874999999998 milliseconds.
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing depth for 800 sites takes 0.10337999999999999 milliseconds.
-----------------------------------------------------------------------------------------------------
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing size for 1600 sites takes 0.11212415999999999 milliseconds.
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing depth for 1600 sites takes 0.11930458999999999 milliseconds.
-----------------------------------------------------------------------------------------------------
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing size for 3200 sites takes 0.36285874999999995 milliseconds.
2021-06-13 19:07:55 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union by storing depth for 3200 sites takes 0.35582333 milliseconds.
```

Evidence to support conclusion

Number of Sites	by Storing Size	by Storing Depth
200	0.102	0.12
400	0.052	0.066
800	0.105	0.103
1600	0.112	0.119
3200	0.363	0.356

Graphical representation



Conclusion

From multiple experiments carried out, It is observed that Weighted Quick Union by storing size takes approximately same run time by storing depth for smaller as well as large number of sites(n). As a result, storing by depth has no effect on run time.

Output

2. With two loops, so that all intermediate nodes point to the root, not just the alternates.

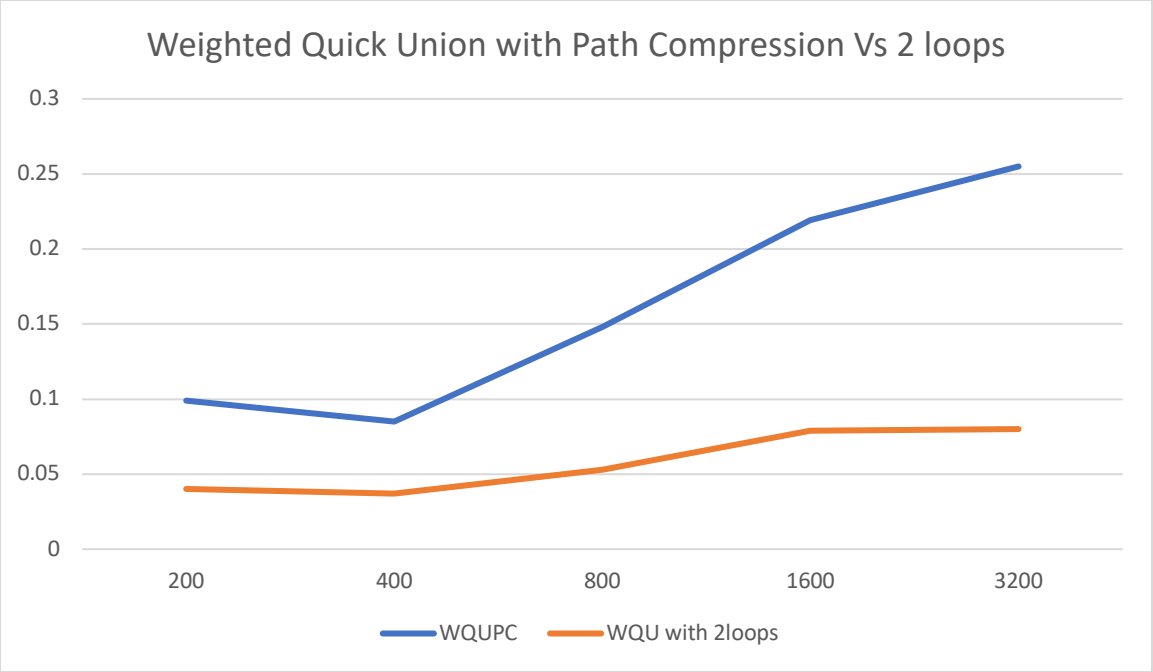
(UF_HWQUPC_alternate2 and UF_HWQUPC_alternate2_benchmark classes added for this part)

```
Run:  UF_HWQUPC_alternate2_benchmark
/Users/sayalimahajan/Library/Java/JavaVirtualMachines/openjdk-15.0.1/Contents/Home/bin/java -javaagent:/Applications/I
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression for 200 sites takes 0.09926625 milliseconds.
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression with 2 loops for 200 sites takes 0.04046208 milliseconds.
-----------------------------------------------------------------------------------------------------
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression for 400 sites takes 0.05514416 milliseconds.
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression with 2 loops for 400 sites takes 0.03730709 milliseconds.
-----------------------------------------------------------------------------------------------------
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression for 800 sites takes 0.10772124999999999 milliseconds.
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression with 2 loops for 800 sites takes 0.05301958 milliseconds.
-----------------------------------------------------------------------------------------------------
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression for 1600 sites takes 0.21915834 milliseconds.
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression with 2 loops for 1600 sites takes 0.07918834 milliseconds.
-----------------------------------------------------------------------------------------------------
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression for 3200 sites takes 0.25464 milliseconds.
2021-06-13 22:03:42 INFO  Benchmark_Timer - Begin run:  with 100 runs
Weighted quick union with path compression with 2 loops for 3200 sites takes 0.08044165999999998 milliseconds.
```

Evidence to support conclusion

Number of sites	WQUPC	WQU with 2loops
200	0.099	0.04
400	0.055	0.037
800	0.108	0.053
1600	0.219	0.079
3200	0.255	0.08

Graphical representation



Conclusion

From multiple experiments carried out, It is observed that When compared to the prior solution, weighted quick union with path compression with two loops increases runtime speed for different sites.