

Program Structures & Algorithms

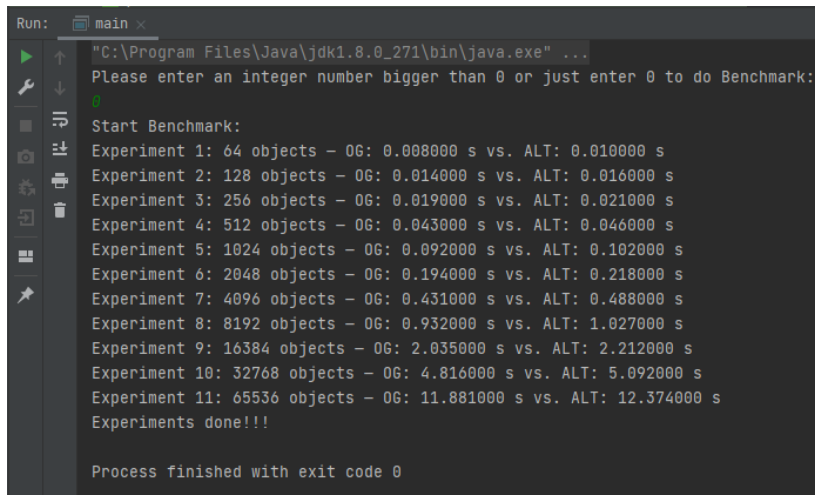
Assignment NO.4

1. Task

- Implement alternative weighted Quick Union using depth (UF_HWQU.java).
- Implement alternative weighted Quick Union with Path Compression using two loops (WQUPC_ALT.java).
- Benchmark the alternative against original implementation (main.java).

2. Output

- Doing experiments (n from 64 to 65536) and 1000-times running of each n:



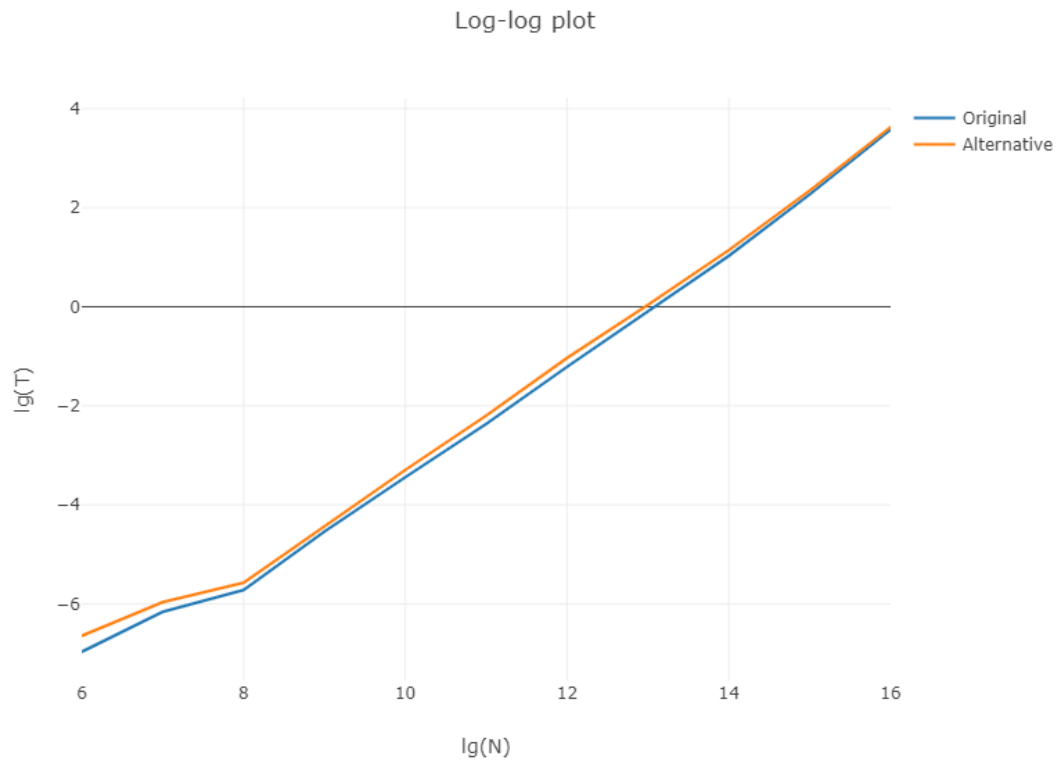
```
Run: main x
"C:\Program Files\Java\jdk1.8.0_271\bin\java.exe" ...
Please enter an integer number bigger than 0 or just enter 0 to do Benchmark:
Start Benchmark:
Experiment 1: 64 objects - OG: 0.008000 s vs. ALT: 0.010000 s
Experiment 2: 128 objects - OG: 0.014000 s vs. ALT: 0.016000 s
Experiment 3: 256 objects - OG: 0.019000 s vs. ALT: 0.021000 s
Experiment 4: 512 objects - OG: 0.043000 s vs. ALT: 0.046000 s
Experiment 5: 1024 objects - OG: 0.092000 s vs. ALT: 0.102000 s
Experiment 6: 2048 objects - OG: 0.194000 s vs. ALT: 0.218000 s
Experiment 7: 4096 objects - OG: 0.431000 s vs. ALT: 0.488000 s
Experiment 8: 8192 objects - OG: 0.932000 s vs. ALT: 1.027000 s
Experiment 9: 16384 objects - OG: 2.035000 s vs. ALT: 2.212000 s
Experiment 10: 32768 objects - OG: 4.816000 s vs. ALT: 5.092000 s
Experiment 11: 65536 objects - OG: 11.881000 s vs. ALT: 12.374000 s
Experiments done!!!
Process finished with exit code 0
```

3. Relationship Conclusion

- Alternative #1 is unnecessary to be benchmarked since no matter what (size or depth) we are using to implement weighted Quick Union, the complexity of find and union are $O(\lg N)$ — they are both making the smaller tree become the child of the larger tree just with different measurement of the tree: size or depth.
- Since there is an additional loop in the find method of alternative weighted Quick Union with Path Compression. The alternative one's complexity of find will add additional $O(\lg N)$ compared with original implementation, but the overall order of growth is still linear.

4. Evidence to support the conclusion:

- Chart



- Table

lg(N)	lg(T)	T	N	categories
6	-6.96578	0.008	64	Original
6	-6.64386	0.01	64	Alternative
7	-6.15843	0.014	128	Original
7	-5.96578	0.016	128	Alternative
8	-5.71786	0.019	256	Original
8	-5.57347	0.021	256	Alternative
9	-4.53952	0.043	512	Original
9	-4.44222	0.046	512	Alternative
10	-3.44222	0.092	1024	Original
10	-3.29336	0.102	1024	Alternative
11	-2.36587	0.194	2048	Original
11	-2.1976	0.218	2048	Alternative
12	-1.21424	0.431	4096	Original
12	-1.03505	0.488	4096	Alternative
13	-0.1016	0.932	8192	Original
13	0.038436	1.027	8192	Alternative

14	1.025029	2.035	16384	Original
14	1.145351	2.212	16384	Alternative
15	2.267835	4.816	32768	Original
15	2.348232	5.092	32768	Alternative
16	3.570584	11.881	65536	Original
16	3.62924	12.374	65536	Alternative

As chart and table shown above, the slope of line in log-log plot indicate that alternative weighted Quick Union with Path Compression has a similar order of growth with original one, that is linear growth. And the results in the table show that original one is faster than alternative one, which is consistent with the conclusion.