Xiaoran Li (NUID: 001023070)

INFO 6205

Program Structures & Algorithms

Fall 2020

Assignment No.4

- **Task**

We mentioned two alternatives for implementing Union-Find:

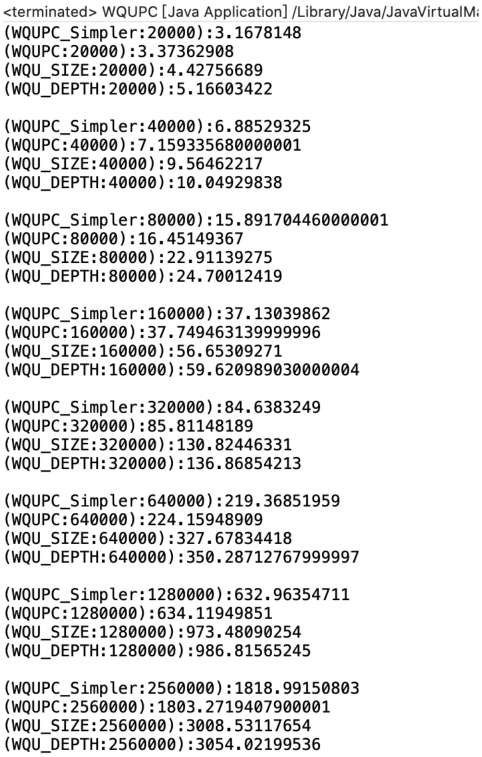
1. For weighted quick union, store the depth rather than the size;

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

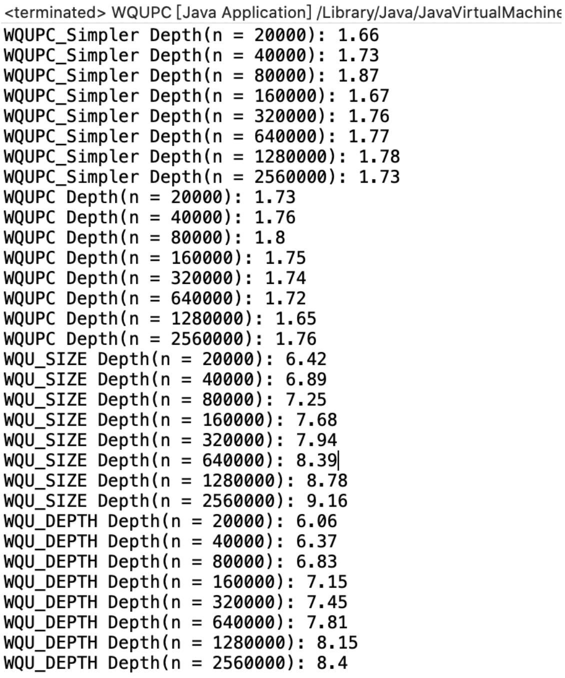
For both of these, code the alternative and benchmark it against the implementation in the repository. You have all of that available from a previous assignment.

If you can explain why alternative #1 is unnecessary to be benchmarked, you may skip benchmarking that one.

- **Output** (few outputs to prove relationship)



```
<terminated> WQUPC [Java Application] /Library/Java/JavaVirtualMa
(WQUPC_Simpler:20000):3.1678148
(WQUPC:20000):3.37362908
(WQU_SIZE:20000):4.42756689
(WQU_DEPTH:20000):5.16603422

(WQUPC_Simpler:40000):6.88529325
(WQUPC:40000):7.159335680000001
(WQU_SIZE:40000):9.56462217
(WQU_DEPTH:40000):10.04929838

(WQUPC_Simpler:80000):15.891704460000001
(WQUPC:80000):16.45149367
(WQU_SIZE:80000):22.91139275
(WQU_DEPTH:80000):24.70012419

(WQUPC_Simpler:160000):37.13039862
(WQUPC:160000):37.749463139999996
(WQU_SIZE:160000):56.65309271
(WQU_DEPTH:160000):59.620989030000004

(WQUPC_Simpler:320000):84.6383249
(WQUPC:320000):85.81148189
(WQU_SIZE:320000):130.82446331
(WQU_DEPTH:320000):136.86854213

(WQUPC_Simpler:640000):219.36851959
(WQUPC:640000):224.15948909
(WQU_SIZE:640000):327.67834418
(WQU_DEPTH:640000):350.28712767999997

(WQUPC_Simpler:1280000):632.96354711
(WQUPC:1280000):634.11949851
(WQU_SIZE:1280000):973.48090254
(WQU_DEPTH:1280000):986.81565245

(WQUPC_Simpler:2560000):1818.99150803
(WQUPC:2560000):1803.2719407900001
(WQU_SIZE:2560000):3008.53117654
(WQU_DEPTH:2560000):3054.02199536
```



```
<terminated> WQUPC [Java Application] /Library/Java/JavaVirtualMachine
WQUPC_Simpler Depth(n = 20000): 1.66
WQUPC_Simpler Depth(n = 40000): 1.73
WQUPC_Simpler Depth(n = 80000): 1.87
WQUPC_Simpler Depth(n = 160000): 1.67
WQUPC_Simpler Depth(n = 320000): 1.76
WQUPC_Simpler Depth(n = 640000): 1.77
WQUPC_Simpler Depth(n = 1280000): 1.78
WQUPC_Simpler Depth(n = 2560000): 1.73
WQUPC Depth(n = 20000): 1.73
WQUPC Depth(n = 40000): 1.76
WQUPC Depth(n = 80000): 1.8
WQUPC Depth(n = 160000): 1.75
WQUPC Depth(n = 320000): 1.74
WQUPC Depth(n = 640000): 1.72
WQUPC Depth(n = 1280000): 1.65
WQUPC Depth(n = 2560000): 1.76
WQU_SIZE Depth(n = 20000): 6.42
WQU_SIZE Depth(n = 40000): 6.89
WQU_SIZE Depth(n = 80000): 7.25
WQU_SIZE Depth(n = 160000): 7.68
WQU_SIZE Depth(n = 320000): 7.94
WQU_SIZE Depth(n = 640000): 8.39
WQU_SIZE Depth(n = 1280000): 8.78
WQU_SIZE Depth(n = 2560000): 9.16
WQU_DEPTH Depth(n = 20000): 6.06
WQU_DEPTH Depth(n = 40000): 6.37
WQU_DEPTH Depth(n = 80000): 6.83
WQU_DEPTH Depth(n = 160000): 7.15
WQU_DEPTH Depth(n = 320000): 7.45
WQU_DEPTH Depth(n = 640000): 7.81
WQU_DEPTH Depth(n = 1280000): 8.15
WQU_DEPTH Depth(n = 2560000): 8.4
```

- **Relationship conclusion**

For weighted quick union, no matter we store the depth or the size, there is no obvious difference in running time(graph 1 in evidence).

For weighted quick union with path compression, no matter we do two loops or implement the grandparent fix, there is no obvious difference in running time(graph 1 in evidence).

For weighted quick union, if we store the depth, the average depth of tree is smaller than the average depth of tree when we store the size.(graph 2 in evidence).

For weighted quick union with path compression, no matter we do two loops or implement the grandparent fix, there is no obvious difference in the average depth of tree(graph 2 in evidence).

The reason why the depth of tree with path compression is much lower than the one without path compression is that WQUPC make every node connect with root directly, while WQU only make node connect with other node randomly.

Furthermore, with the growth of the number of objects, for WQU without PC, the average depth of tree is higher and the running time will even higher.

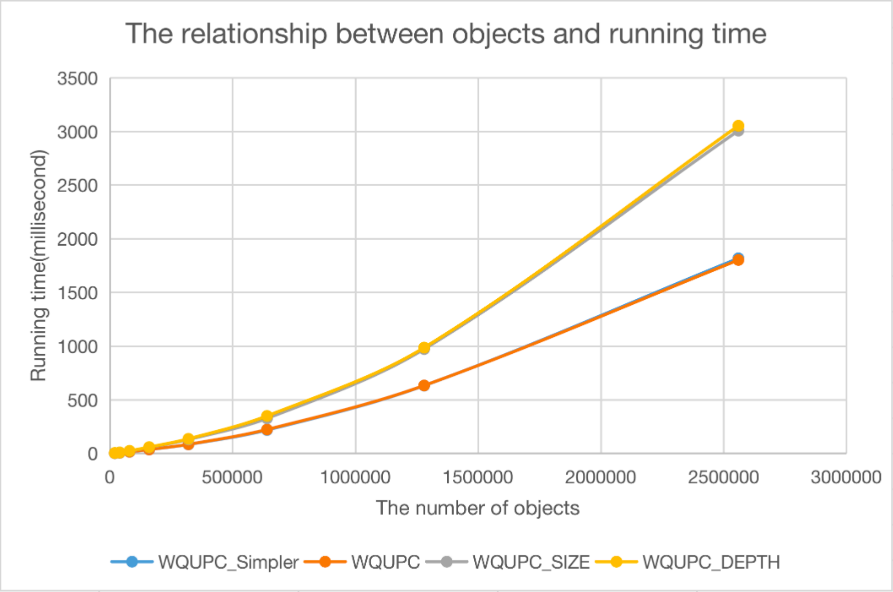
For WQUPC, if we do two loops, the result is that we get a tree that every node is connected to root. If we implement the grandparent fix, it seems like we only make every node points to its grandparent, but we can easily found that every node's parent is root. It means every node points to parent[root]. Therefore, the performance of these two alternative implementations are similar.

For WQU, if we store the depth, it is sure that every lower tree is connected to higher tree. However, if we store the size, sometimes a higher tree with small size will be connected to a lower tree with large size. This situation causes the different depth of tree.

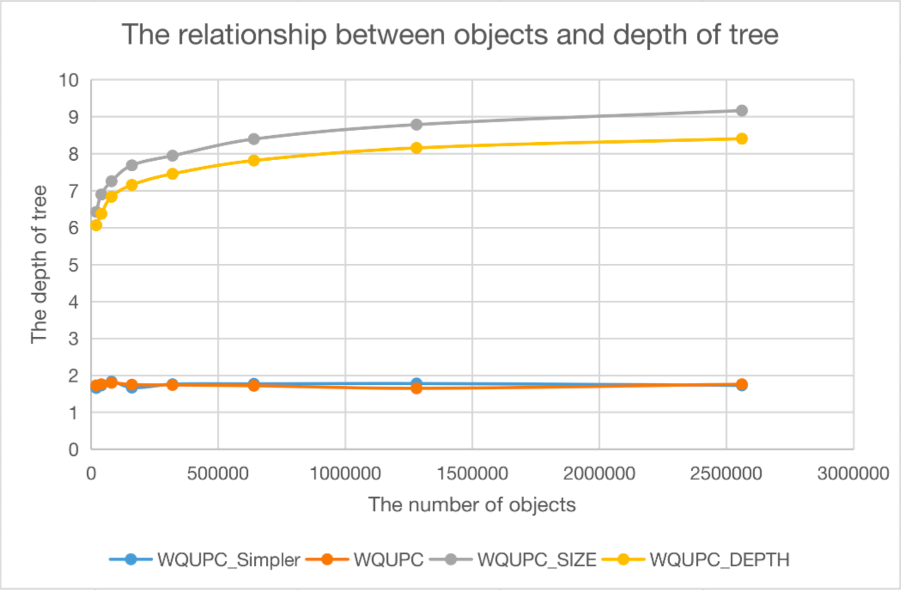
- **Evidence to support relationship** (screen shot and/or graph and/or spreadsheet)

Xiaoran Li (NUID: 001023070)

n	WQUPC_Simpler	WQUPC	WQUPC_SIZE	WQUPC_DEPTH
20000	3.1678148	3.37362908	4.42756689	5.16603422
40000	6.88529325	7.15933568	9.56462217	10.04929838
80000	15.89170446	16.45149367	22.91139275	24.70012419
160000	37.13039862	37.74946314	56.65309271	59.62098903
320000	84.6383249	85.81148189	130.8244633	136.8685421
640000	219.3685196	224.1594891	327.6783442	350.2871277
1280000	632.9635471	634.1194985	973.4809025	986.8156525
2560000	1818.991508	1803.271941	3008.531177	3054.021995



n	WQUPC_Simpler	WQUPC	WQUPC_SIZE	WQUPC_DEPTH
20000	1.66	1.73	6.42	6.06
40000	1.73	1.76	6.89	6.37
80000	1.83	1.8	7.25	6.83
160000	1.67	1.75	7.68	7.15
320000	1.76	1.74	7.94	7.45
640000	1.77	1.72	8.39	7.81
1280000	1.78	1.65	8.78	8.15
2560000	1.73	1.76	9.16	8.4


The relationship between objects and depth of tree
The depth of tree
The number of objects
WQUPC_Simpler
WQUPC
WQUPC_SIZE
WQUPC_DEPTH