**INFO 6205**

**Program Structures & Algorithms**

**Fall 2020**

**Assignment No 4**

* **Task**

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

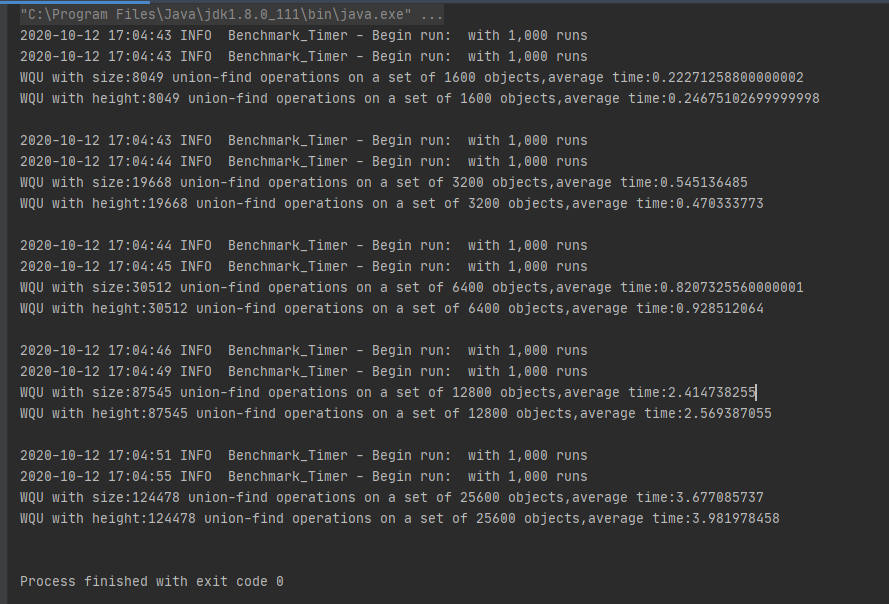
For weighted quick union with path compression, do two loops, so that all intermediate nodes point to the root, not just the alternates.

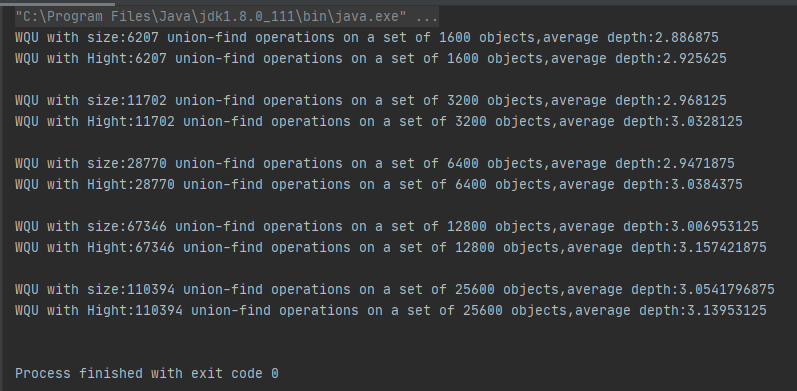
For both of these, code the alternative and benchmark it against the implementation in the repository. You have all of that available from a previous assignment.

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

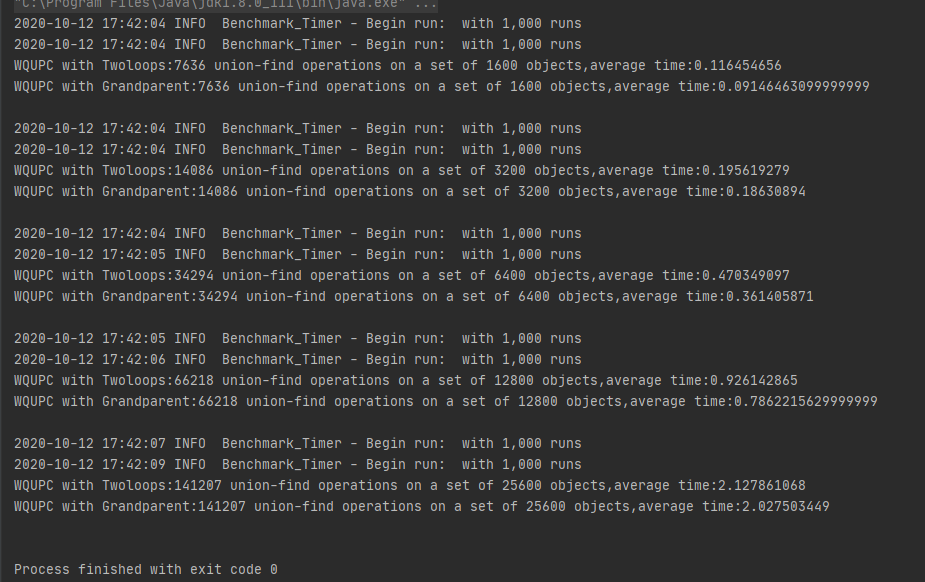
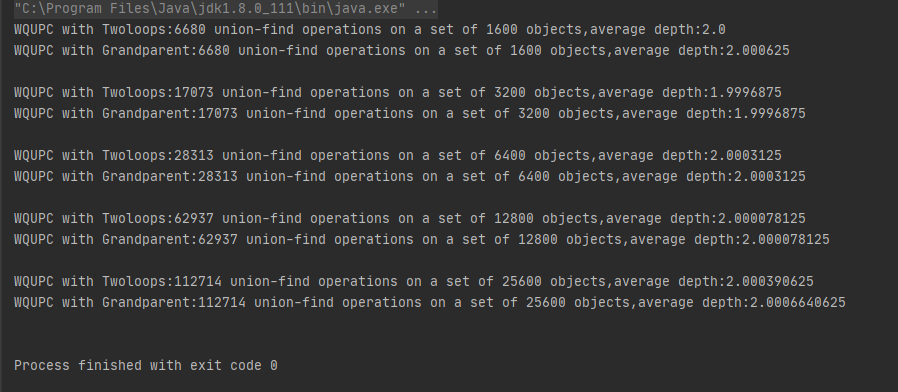
For a set of N objects which N={1600,3200,6400,12800,25600},random union them till they are all connected.

For question 1





### For question 2



* **Relationship conclusion**

1. For weighted quick union, store the height or depth rather than the size would make average depth bigger,and cost more time.(because time is related to the depth)
2. For weighted quick union with path compression, do two loops rather than Simpler one-pass variant would cost more time during find();because their average depth are almost same.

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

In order to complete this experiment, I copy WQUPC.class and simple fix it as WQU\_Hight.class,WQU\_Size.class and WQUPC\_grandparent.class;

WQU\_Hight.class and WQU\_Size.class used for question1 in main method of UExperiment;

WQUPC\_grandparent.class and WQUPC used for question2 in;



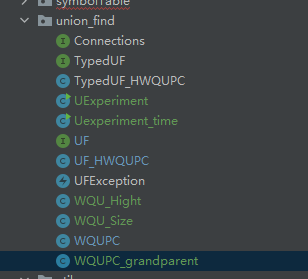


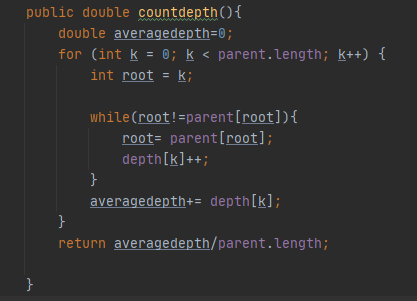
UExperiment.class used for compare average depth;

UExperiment\_time.class used for compare average run time;

Beside, I add a method called countdepth() in WQUPCWQU\_Hight.class,WQU\_Size.class and WQUPC\_grandparent.class to count depth.

All of these class are under union\_find Package.







* **Screenshot of Unit test passing**

