Kshitij Verma NUID: 001305285

INFO 6205

Program Structures & Algorithms

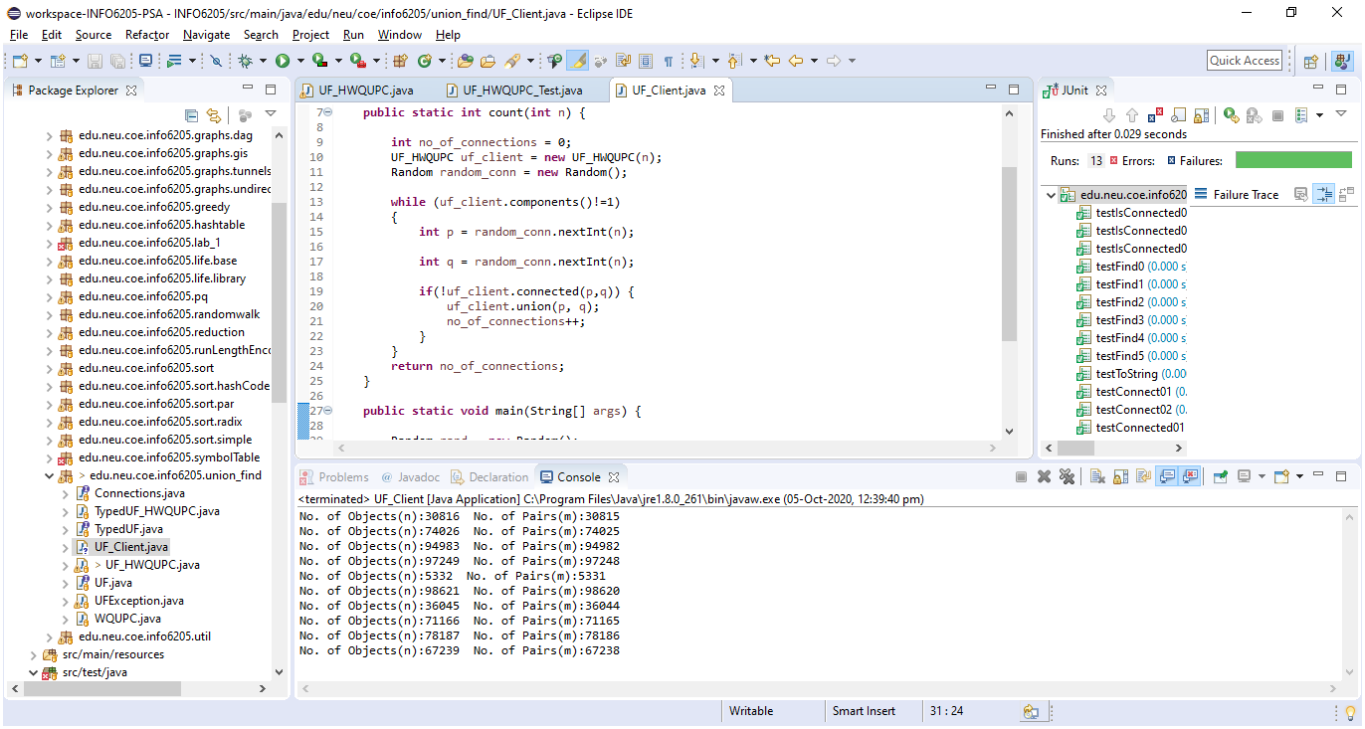
Fall 2020

Assignment No: 3

- **Task:** To implement height-weighted Quick union with Path compression. For this task UF_HWQUPC java class was used and following methods were implemented-
    - find () – method to update the root of input object if pathCompression is performed
    - mergeComponents() – method to merge 2 subtrees such that smaller root points to larger root
    - doPathCompression() – method that implements the single-pass process of pathCompression method.

  Also, UF_Client java class was devised to perform and test the implementation of UF_HWQUPC class

- **Output:** Below are the values of (m) pairs generated for 10 different values of objects (n) as input -



Console Output-

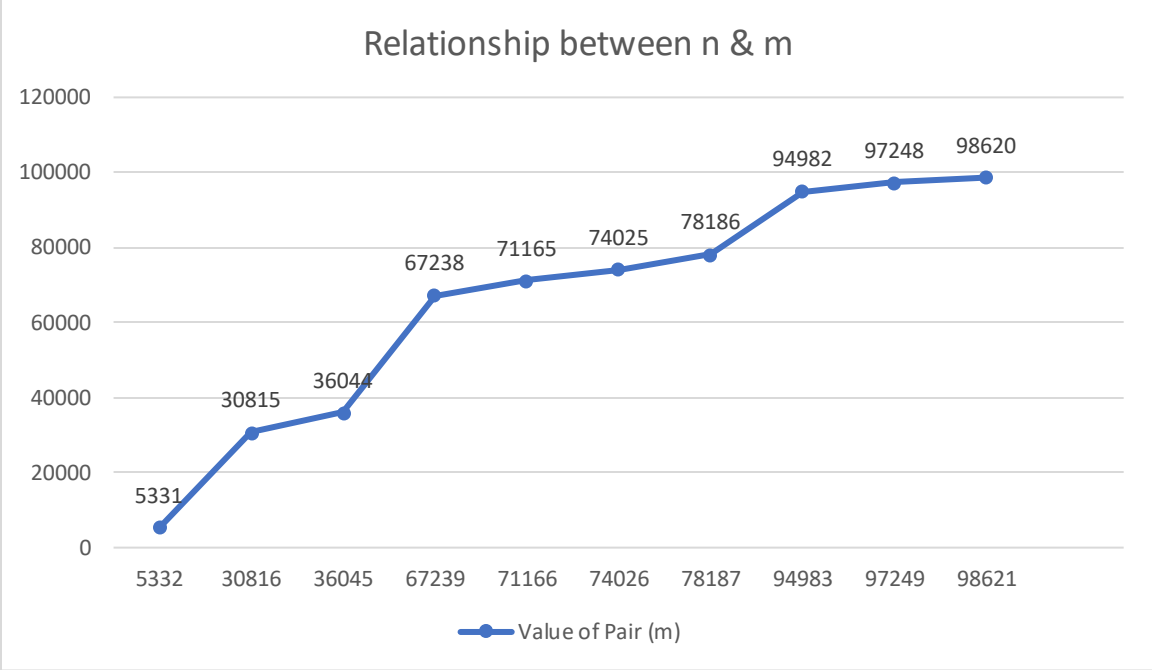
```
No. of Objects(n):30816  No. of Pairs(m):30815
No. of Objects(n):74026  No. of Pairs(m):74025
No. of Objects(n):94983  No. of Pairs(m):94982
No. of Objects(n):97249  No. of Pairs(m):97248
No. of Objects(n):5332  No. of Pairs(m):5331
No. of Objects(n):98621  No. of Pairs(m):98620
No. of Objects(n):36045  No. of Pairs(m):36044
No. of Objects(n):71166  No. of Pairs(m):71165
No. of Objects(n):78187  No. of Pairs(m):78186
No. of Objects(n):67239  No. of Pairs(m):67238
```

- **Relationship conclusion:** It can be concluded from the results mentioned above that the number of pairs(m) generated are proportional to the number of objects provided as input.

  i.e. $m = n - 1$

- **Evidence to support relationship:** I have attached a chart and a table stating the data of the different output (m) observed for the different set of inputs of n. As a result, we can see the proportionate result between the values of n and m-

Value of Object (n)	Value of Pair (m) generated
30816	30815
74026	74025
94983	94982
97249	97248
5332	5331
98621	98620
36045	36044
71166	71165
78187	78186
67239	67238



- **Screenshot of Unit test passing:** Below is the screenshot of all the unit tests which ran successfully

UF_HWQUPC_Test