**Jyoti Vaswani (NUID -001005711)**

**INFO 6205**

**Program Structures and Algorithms**

**Fall 2020**

**Assignment No -3**

**Note: If any of the inserted object/file fails to open, kindly check the zip. I have added all the supporting files.**

**Task :**

1. **I have added the code in UF\_HWQUPC.java for the methods listed below:**

* **find () –**updates the root of the input site based on whether pathCompression is to be performed or not.
* **mergeComponents ()-**mergers the 2 subtrees in a way that makes smaller root point to larger one.
* **doPathCompression()** - implements the single-pass path-halving mechanism of path compression

1. **I have UF\_Client.java with the static methods listed below:**

* **main () –** Generates random value for n which is taken as an input for finding its corresponding number of connections required to achieve a complete union.
* **count ()-** generates random pairs of integers within the range 0 to n-1 for creating connections and returns the count of connections required for achieving a complete union of n input sites.

** **

1. **Output:**

Below are the screenshots of output generated by **UF\_Client**.java over 10 iterations for different input values of n. (number of sites -n).

‘n’ takes random value in all the iterations

**  **

1. **Relationship Conclusion:**

The number of pairs (m) generated are proportional to number of sites (n) taken as input.

The equation describing the relation is :

**m= n-1**

Refer the attached output screenshots as evidence.

**  **

1. **Evidence to support relationship:**

To support the relationship described above, I took 10 different values of n (generated randomly) – in 3 runs.

All the observations are added in below screenshots.

**  **

1. **Screenshot of Unit test passing:**

No change was made in any of the test.java files.

Below are the attached screenshots of successful execution and results.

