**UF\_HWQUPC Assignemnt**

**Step 1:** Union(merge) and find methods are implemented along with path compression

**Union Find**

**Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

**Step 1(b):** All the test cases are passed successfully

**Graphical user interface, text, application

Description automatically generated**

**Step 2:** Implemented a count() method in HWQUPC\_Solution class that generates random pairs of integers, union them if they are not connected, does the path compression, and returns the no of connections generated.

Graphical user interface, text, application

Description automatically generated

*Also implemented a main method in the UF\_HWQUPC class that takes manual inputs during run time and unions the objects, returns the connected objects and, prints number of connections when everything is connected to check if the union find is working.*

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**Step 3:** Implemented a main function that runs the count method a number of times and prints the average of the number of connections generated.

**n** is no.of objects, **trials** is no.of times the program is run. For each value of n, program is run for **trials** no of times and prints the average no of connections generated.

**Graphical user interface, text, application

Description automatically generated**

After running that **for sufficient number of objects(sufficiently large)** for **1000 trials** each and average is calculated, plotted a graph for objects vs connections. The plot almost looks like a straight line. The average **number of connections** generated is equal to **n\*ln(n)/2** where n is number of random objects.

Chart, scatter chart

Description automatically generated