

Tanay Saxena (001586302)
Program Structures & Algorithms
Fall 2021
Assignment No. 3

- Task (List down the tasks performed in the Assignment)
 - Added missing code in 4 methods -
Class UF_HWQUPC.java
find() -

```
public int find(int p) {  
    validate(p);  
    int root = p;  
    // TO BE IMPLEMENTED  
    if (pathCompression) doPathCompression(root);  
    while (root != parent[root]) {  
        root = parent[root];  
    }  
    return root;  
}
```

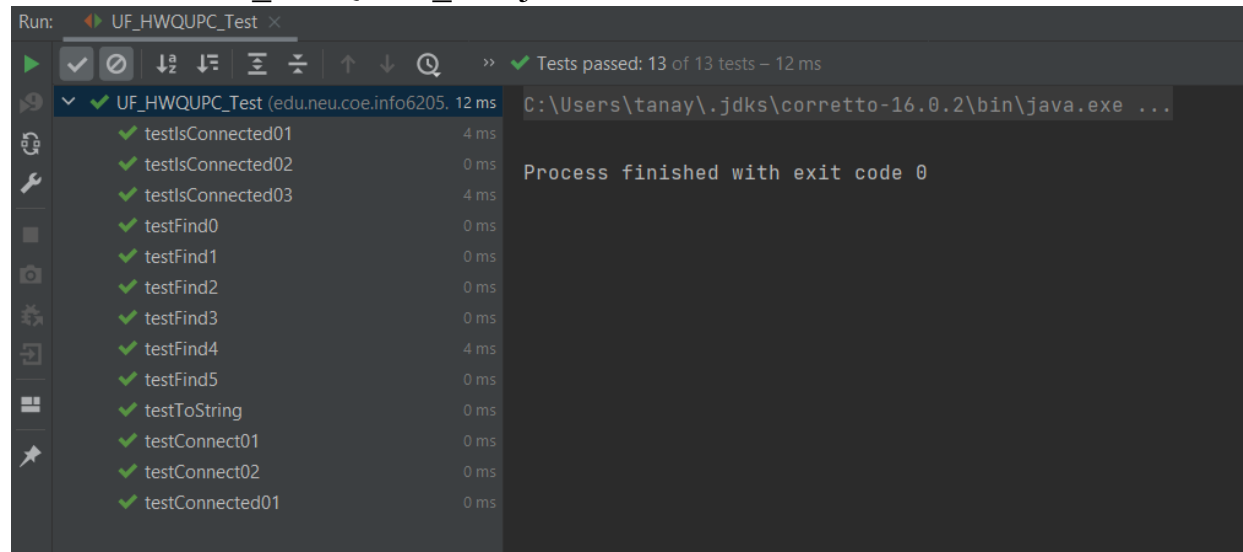
mergeComponents() -

```
private void mergeComponents(int i, int j) {  
    // TO BE IMPLEMENTED make shorter root point to taller one  
    if (i == j) return;  
    if (height[i] < height[j]) {  
        updateParent(i, j);  
    }  
    else {  
        updateParent(j, i);  
    }  
    if (height[i] == height[j]) {  
        height[i]++;  
    }  
}
```

doPathCompression()

```
private void doPathCompression(int i) {  
    // TO BE IMPLEMENTED update parent to value of grandparent  
    while (i != parent[i]) {  
        parent[i] = parent[parent[i]];  
        i = parent[i];  
    }  
}
```

- Ran tests for UF_HWQUPC_Test.java-



Run: UF_HWQUPC_Test x

Tests passed: 13 of 13 tests – 12 ms

Process finished with exit code 0

Test Name	Duration
testIsConnected01	4 ms
testIsConnected02	0 ms
testIsConnected03	4 ms
testFind0	0 ms
testFind1	0 ms
testFind2	0 ms
testFind3	0 ms
testFind4	4 ms
testFind5	0 ms
testToString	0 ms
testConnect01	0 ms
testConnect02	0 ms
testConnected01	0 ms

- Created a new class UnionFindClient.java for the analysis -

```
10 public class UnionFindClient {  
11  
12     public static double count(int n, int epochs) {  
13         double avgCount = 0;  
14         for (int t = 0; t < epochs; t++) {  
15             UF_HWQUPC dsu = new UF_HWQUPC(n, pathCompression: true);  
16             Random random = new Random();  
17             int curCount = 0;  
18             while (dsu.components() != 1) {  
19                 dsu.connect((int)(Math.random() * n), (int)(Math.random() * n));  
20                 curCount++;  
21             }  
22             avgCount += curCount;  
23         }  
24         return avgCount/epochs;  
25     }  
}
```

```

26
27 public static void main(String[] args) {
28     try {
29         String fileSeparator = File.separator;
30         FileWriter writer = new FileWriter(Paths.get( first: "assignment_reports",
31             ...more: "assignment3_Tanay_Saxena", "union_find.csv").toString());
32         writer.write( str: "inputSize,pairCount\n");
33         for (int n = 64; n <= 10000; n += 30) {
34             double m = UnionFindClient.count(n, epochs: 10);
35             System.out.println("Number of steps n = " + n + ", number of unions required m = " + m);
36             writer.write( str: n + ", " + m + "\n");
37         }
38         writer.close();
39     }
40     catch (IOException e) {
41         e.printStackTrace();
42     }
43 }
44 }

```

- Generated the output by executing the main method -

```

C:\Users\tanay\.jdk\corretto-16.0.2\bin\java.exe ...
Number of steps n = 64, number of unions required m = 160.6
Number of steps n = 94, number of unions required m = 235.4
Number of steps n = 124, number of unions required m = 324.5
Number of steps n = 154, number of unions required m = 449.4
Number of steps n = 184, number of unions required m = 525.0
Number of steps n = 214, number of unions required m = 638.2
Number of steps n = 244, number of unions required m = 651.8
Number of steps n = 274, number of unions required m = 836.9
Number of steps n = 304, number of unions required m = 999.8
Number of steps n = 334, number of unions required m = 963.6
Number of steps n = 364, number of unions required m = 1147.3
Number of steps n = 394, number of unions required m = 1351.5
Number of steps n = 424, number of unions required m = 1483.7
Number of steps n = 454, number of unions required m = 1590.9
Number of steps n = 484, number of unions required m = 1734.1
Number of steps n = 514, number of unions required m = 1644.6

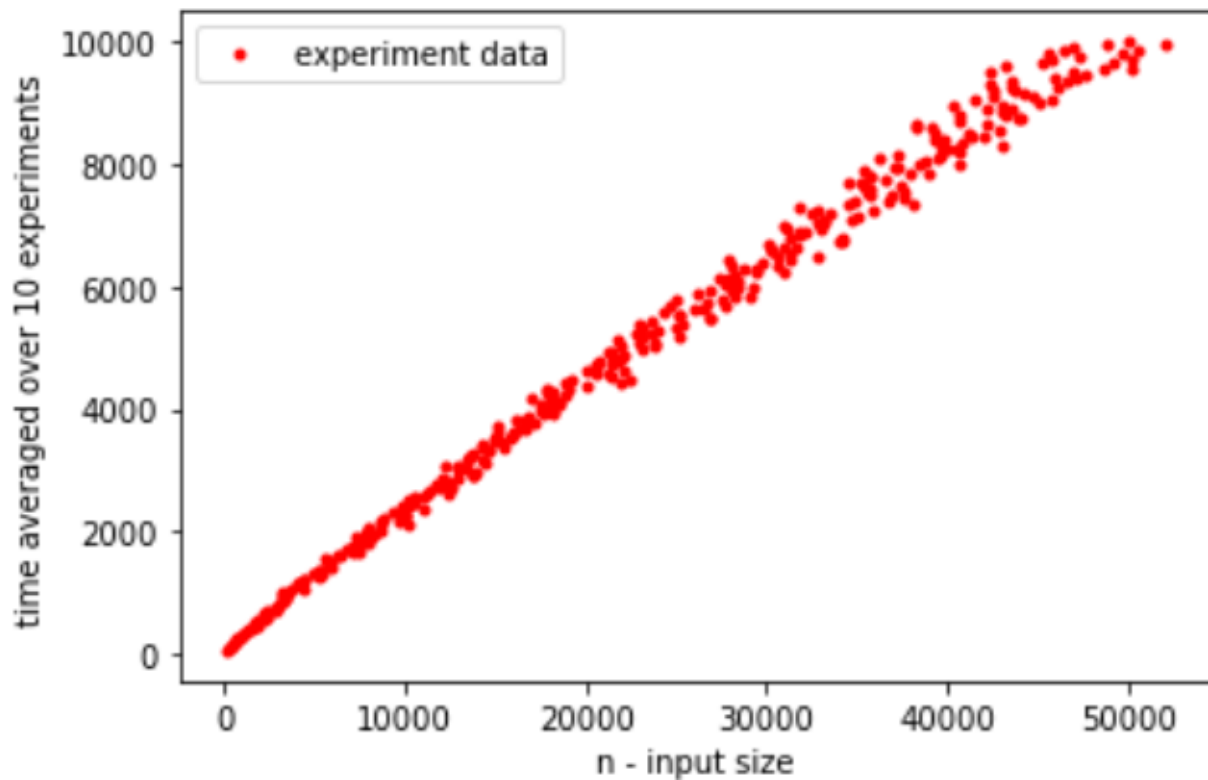
```

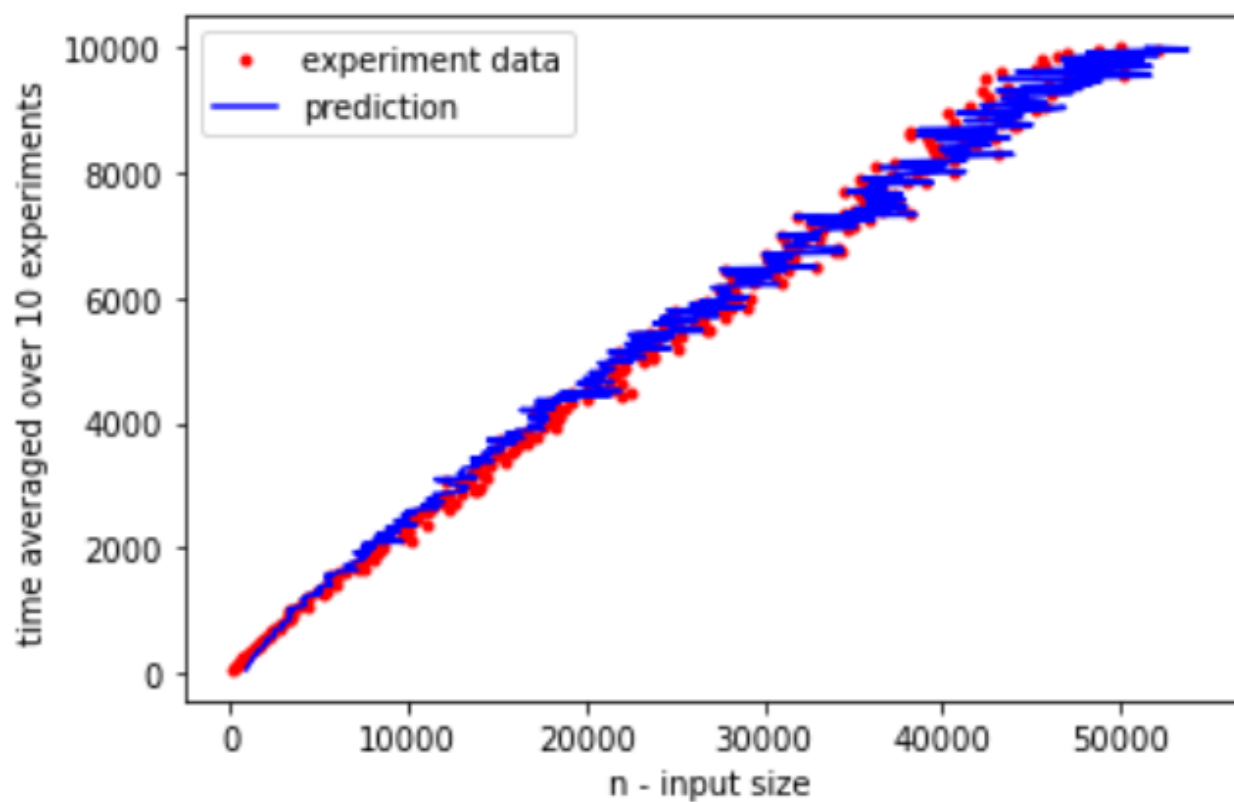
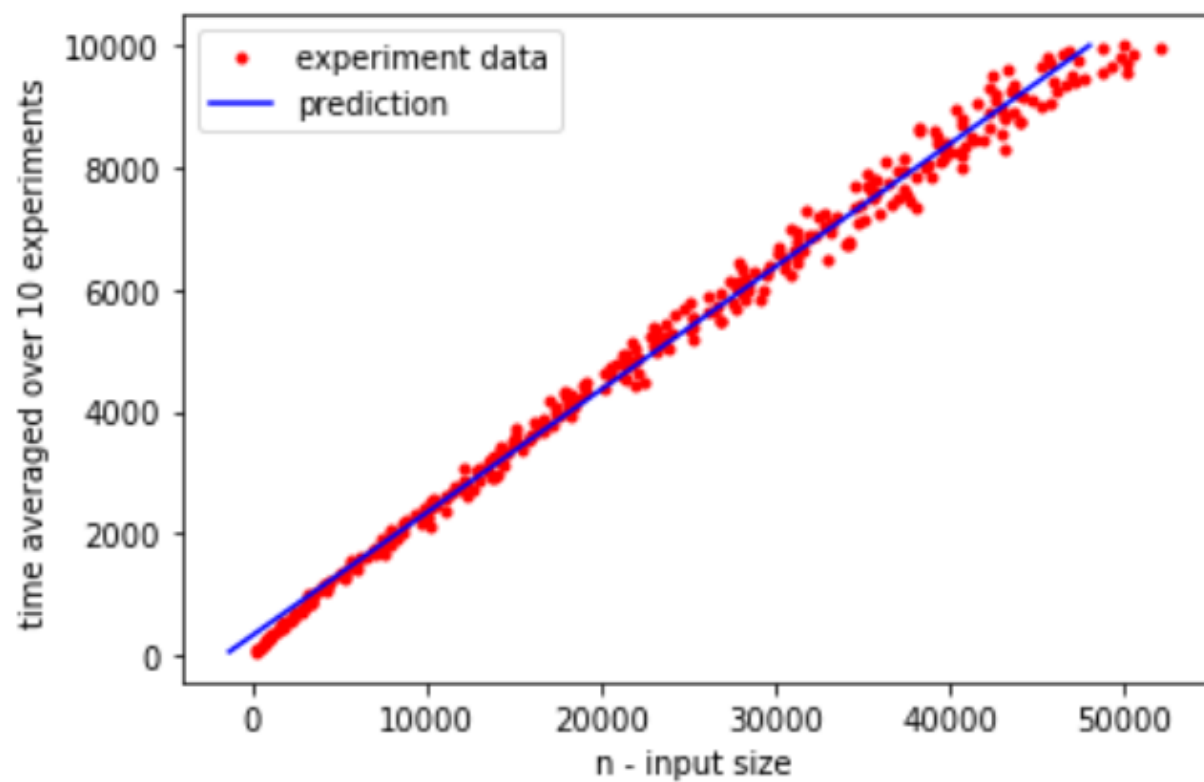
.....

```
Number of steps n = 9694, number of unions required m = 50195.4
Number of steps n = 9724, number of unions required m = 45784.3
Number of steps n = 9754, number of unions required m = 47330.9
Number of steps n = 9784, number of unions required m = 49748.8
Number of steps n = 9814, number of unions required m = 45638.3
Number of steps n = 9844, number of unions required m = 46525.4
Number of steps n = 9874, number of unions required m = 50509.1
Number of steps n = 9904, number of unions required m = 46921.9
Number of steps n = 9934, number of unions required m = 48843.1
Number of steps n = 9964, number of unions required m = 52037.7
Number of steps n = 9994, number of unions required m = 49968.8

Process finished with exit code 0
```

- Created a jupyter notebook and performed analysis on the output (csv format)





- Relationship Conclusion:
 - The graph appears to be linearithmic,
 $M = a * N \lg(N) + b$
 - N: input size
 - M: average number of pairs
 - lg: log base 2
 - a & b are constants
 - The detailed experiment can be found [here](#)

- Evidence to support the conclusion:

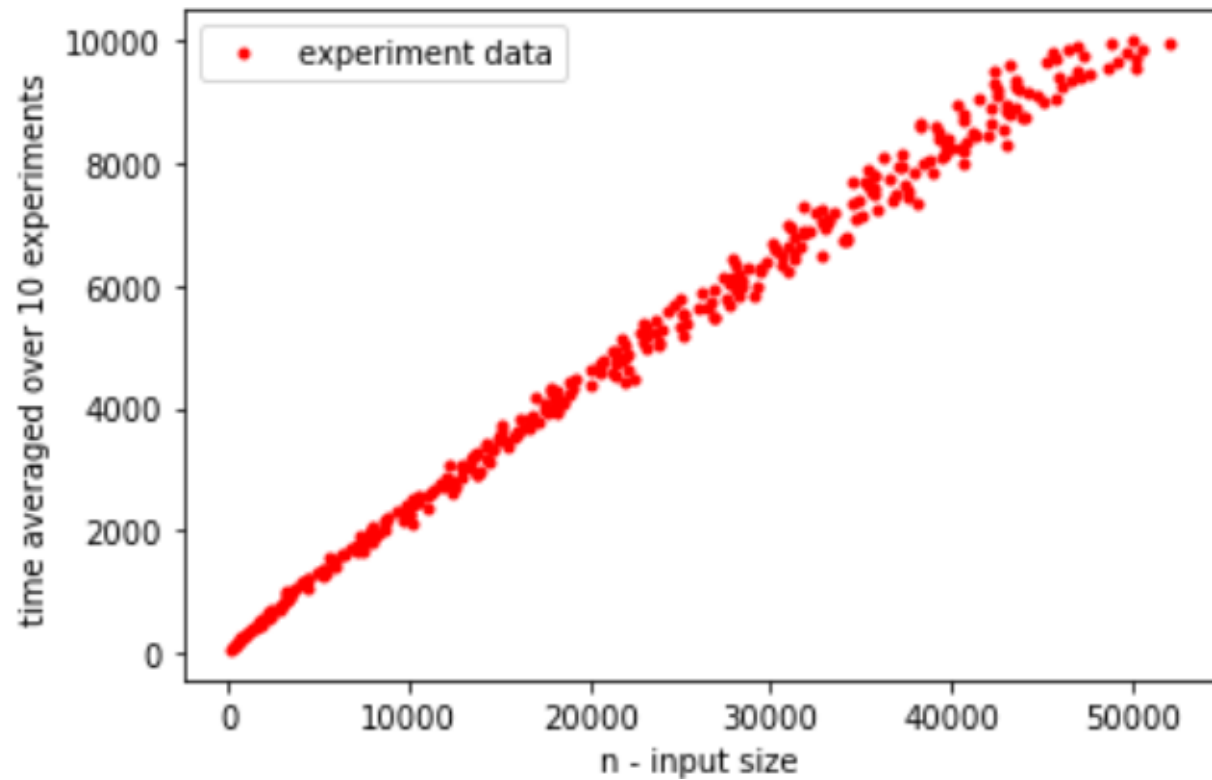
1. Output (Snapshot of Code output in the terminal)

```
C:\Users\tanay\.jdk\corretto-16.0.2\bin\java.exe ...  
Number of steps n = 64, number of unions required m = 160.6  
Number of steps n = 94, number of unions required m = 235.4  
Number of steps n = 124, number of unions required m = 324.5  
Number of steps n = 154, number of unions required m = 449.4  
Number of steps n = 184, number of unions required m = 525.0  
Number of steps n = 214, number of unions required m = 638.2  
Number of steps n = 244, number of unions required m = 651.8  
Number of steps n = 274, number of unions required m = 836.9  
Number of steps n = 304, number of unions required m = 999.8  
Number of steps n = 334, number of unions required m = 963.6  
Number of steps n = 364, number of unions required m = 1147.3  
Number of steps n = 394, number of unions required m = 1351.5  
Number of steps n = 424, number of unions required m = 1483.7  
Number of steps n = 454, number of unions required m = 1590.9  
Number of steps n = 484, number of unions required m = 1734.1  
Number of steps n = 514, number of unions required m = 1644.6
```

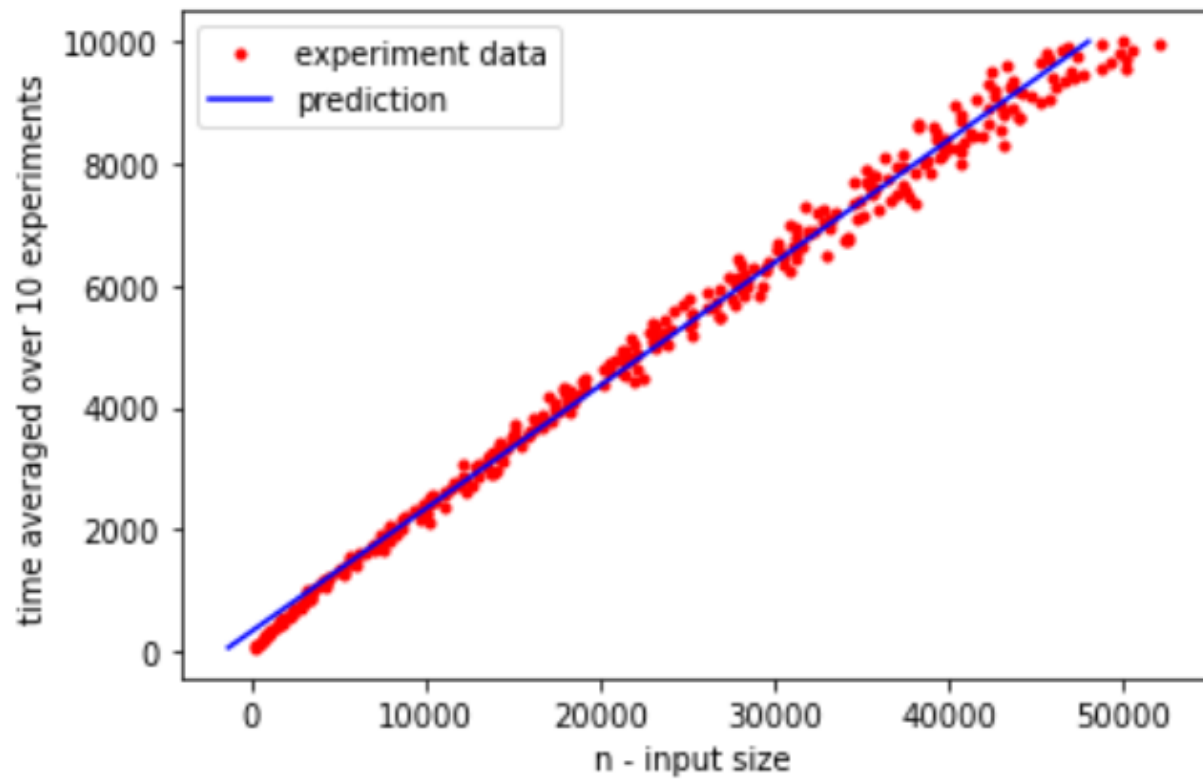
```
Number of steps n = 9694, number of unions required m = 50195.4
Number of steps n = 9724, number of unions required m = 45784.3
Number of steps n = 9754, number of unions required m = 47330.9
Number of steps n = 9784, number of unions required m = 49748.8
Number of steps n = 9814, number of unions required m = 45638.3
Number of steps n = 9844, number of unions required m = 46525.4
Number of steps n = 9874, number of unions required m = 50509.1
Number of steps n = 9904, number of unions required m = 46921.9
Number of steps n = 9934, number of unions required m = 48843.1
Number of steps n = 9964, number of unions required m = 52037.7
Number of steps n = 9994, number of unions required m = 49968.8

Process finished with exit code 0
```

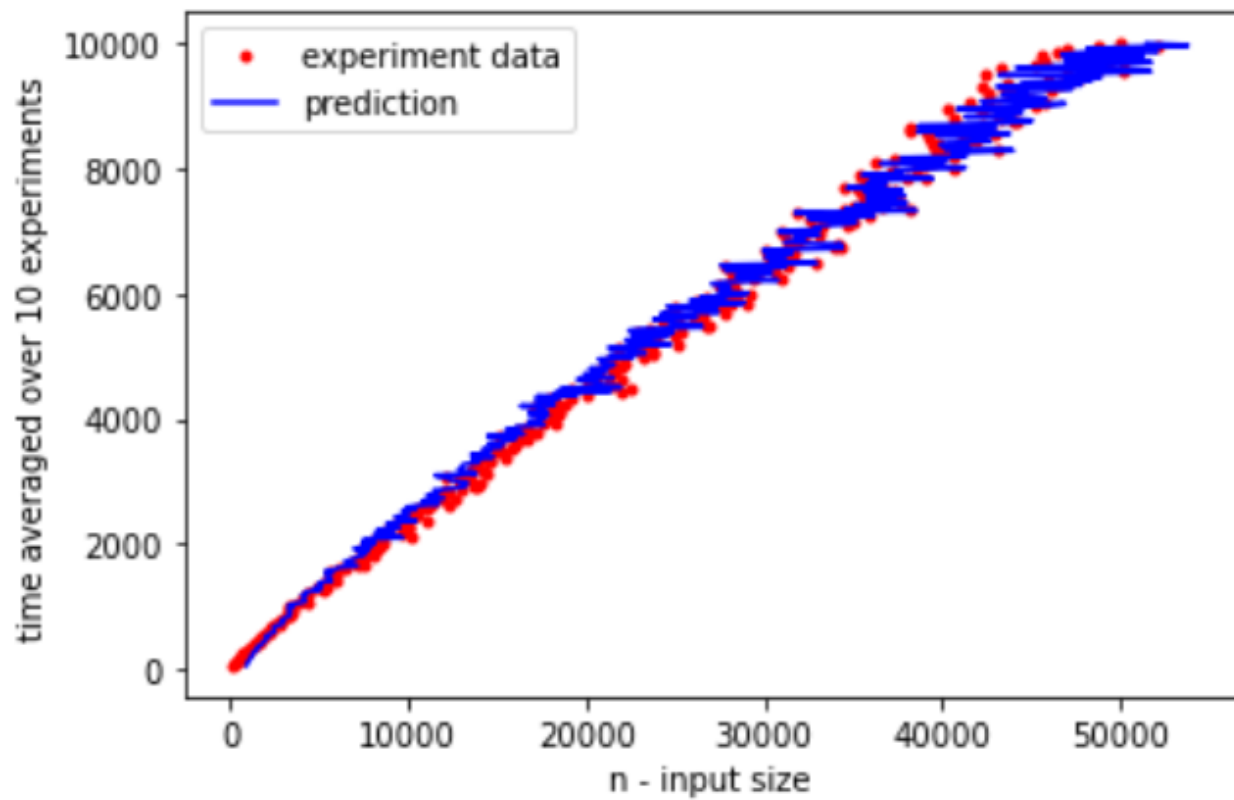
2. Graphical Representation(Observations)



○



○



○

3. Unit tests result:(Snapshot of successful unit test run)

Run: UF_HWQUPC_Test ×

✓ Tests passed: 13 of 13 tests – 12 ms

✓ UF_HWQUPC_Test (edu.neu.coe.info6205, 12 ms) C:\Users\tanay\.jdk\corretto-16.0.2\bin\java.exe ...

✓ testIsConnected01 4 ms

✓ testIsConnected02 0 ms

✓ testIsConnected03 4 ms

✓ testFind0 0 ms

✓ testFind1 0 ms

✓ testFind2 0 ms

✓ testFind3 0 ms

✓ testFind4 4 ms

✓ testFind5 0 ms

✓ testToString 0 ms

✓ testConnect01 0 ms

✓ testConnect02 0 ms

✓ testConnected01 0 ms

Process finished with exit code 0