

Program Structures & Algorithms

Spring 2022

Assignment No. 3

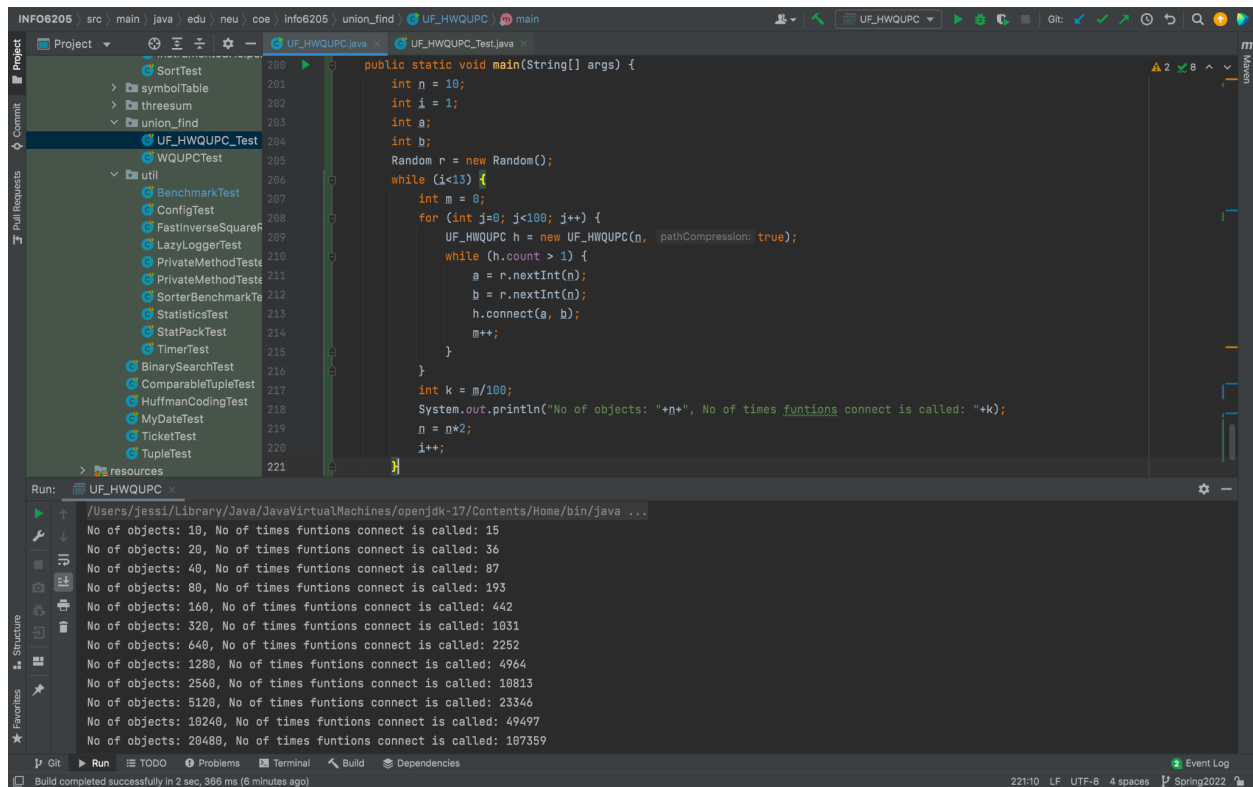
Name: Jashwanth Reddy Kamsani

(NUID): 002988299

Task:

- 1) (a) Implement height-weighted Quick Union with Path Compression.
(b) Check that the unit tests for this class all work.
- 2) Create a main program that takes n from the command line, calls `count()` and prints the returned value.
- 3) Determine the relationship between the number of objects (n) and the number of pairs (m)

Output screenshot:



The screenshot shows an IDE with a project named 'UF_HWQUPC'. The left sidebar displays a package explorer with a tree structure including 'SortTest', 'symbolTable', 'threesum', 'union_find', 'UF_HWQUPC_Test', 'WQUPCTest', 'util', 'BenchmarkTest', 'ConfigTest', 'FastInverseSquare', 'LazyLoggerTest', 'PrivateMethodTest', 'PrivateMethodTeste', 'SorterBenchmarkTe', 'StatisticsTest', 'StatPackTest', 'TimerTest', 'BinarySearchTest', 'ComparableTupleTest', 'HuffmanCodingTest', 'MyDataTest', 'TicketTest', and 'TupleTest'. The main editor displays the 'UF_HWQUPC.java' file with the following code:

```
public static void main(String[] args) {
    int n = 10;
    int i = 1;
    int a;
    int b;
    Random r = new Random();
    while (i < 13) {
        int m = 0;
        for (int j = 0; j < 100; j++) {
            UF_HWQUPC h = new UF_HWQUPC(n, pathCompression: true);
            while (h.count > 1) {
                a = r.nextInt(n);
                b = r.nextInt(n);
                h.connect(a, b);
                m++;
            }
        }
        int k = m / 100;
        System.out.println("No of objects: " + n + ", No of times funtions connect is called: " + k);
        n = n * 2;
        i++;
    }
}
```

The bottom panel shows the 'Run' output for 'UF_HWQUPC' with the following text:

```
/Users/jessi/Library/Java/JavaVirtualMachines/openjdk-17/Contents/Home/bin/java ...
No of objects: 10, No of times funtions connect is called: 15
No of objects: 20, No of times funtions connect is called: 36
No of objects: 40, No of times funtions connect is called: 87
No of objects: 80, No of times funtions connect is called: 193
No of objects: 160, No of times funtions connect is called: 442
No of objects: 320, No of times funtions connect is called: 1031
No of objects: 640, No of times funtions connect is called: 2252
No of objects: 1280, No of times funtions connect is called: 4964
No of objects: 2560, No of times funtions connect is called: 10813
No of objects: 5120, No of times funtions connect is called: 23346
No of objects: 10240, No of times funtions connect is called: 49497
No of objects: 20480, No of times funtions connect is called: 107359
```

Conclusion:

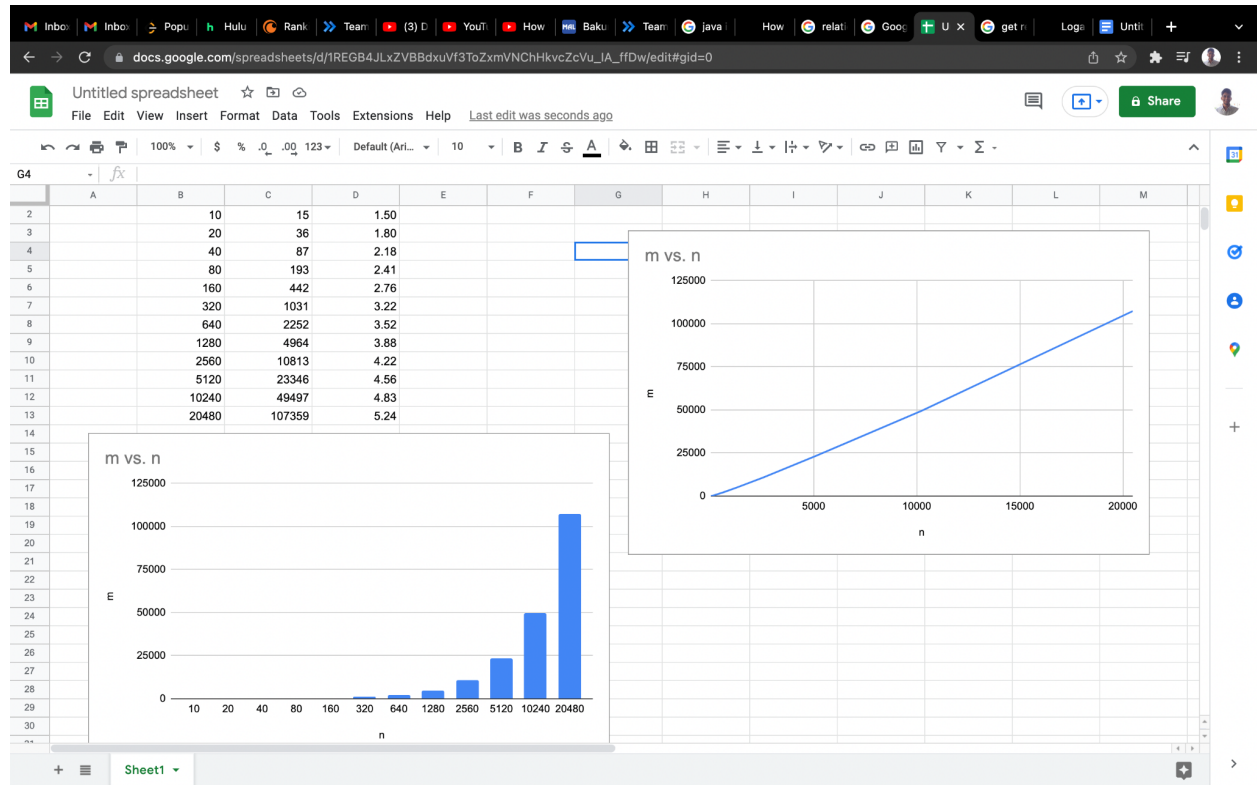
As the number of objects(n) are increasing the number of pairs(m) increasing in n times the logarithm of n with base approximately 5.

$$m = n * \log_5(n)$$

For larger values of n which is around 5000 to 20000, m value is nearly 5 times of n which can be seen in the evidence graph.

$$m = 5n$$

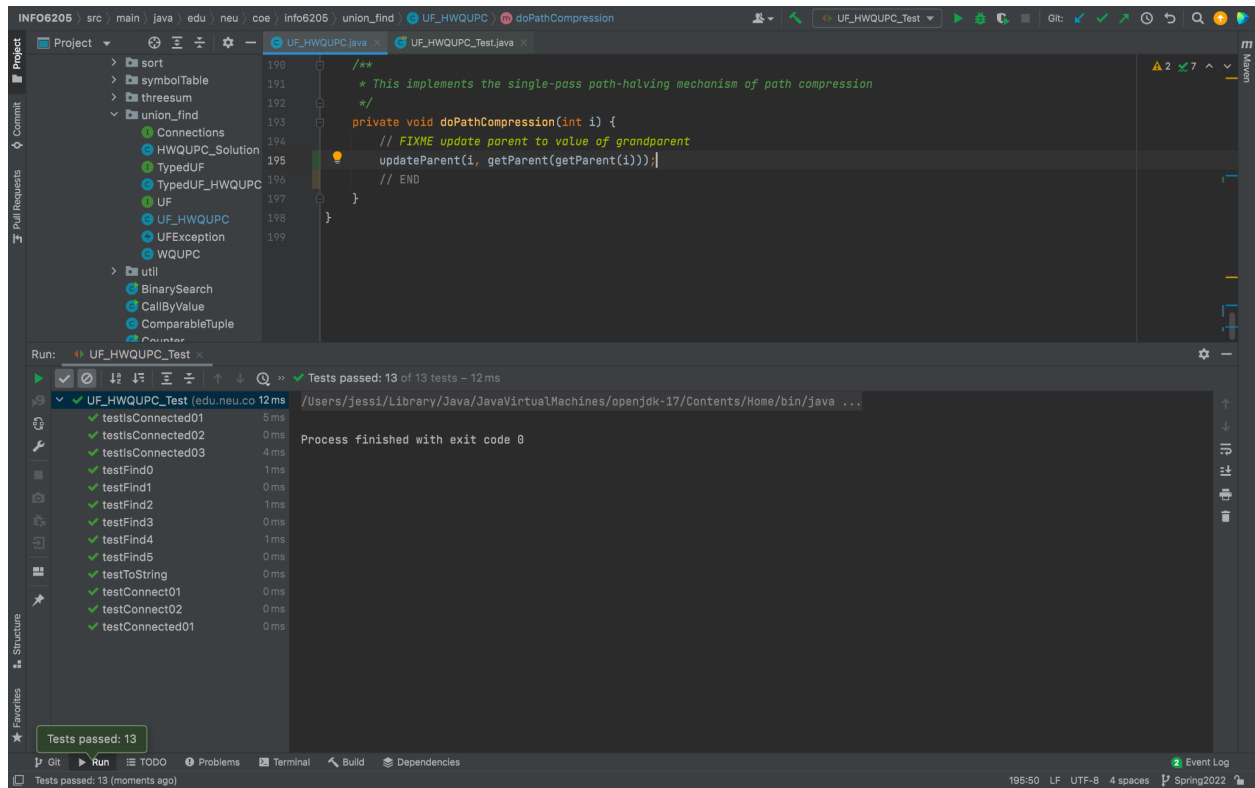
Evidence / Graph:



n	m	m/n
10	15	1.50
20	36	1.80
40	87	2.18
80	193	2.41
160	442	2.76
320	1031	3.22
640	2252	3.52
1280	4964	3.88
2560	10813	4.22
5120	23346	4.56
10240	49497	4.83
20480	107359	5.24

Unit tests result

Part 1:



The screenshot shows an IDE with the following components:

- Project Explorer:** Displays a project structure with folders like 'sort', 'symbolTable', 'threesum', 'union_find', and 'util'. The 'union_find' folder is expanded, showing files like 'Connections', 'HWQUPC_Solution', 'TypedUF', 'TypedUF_HWQUPC', 'UF', 'UF_HWQUPC', 'UFException', and 'WQUPC'.
- Code Editor:** Shows the implementation of the 'doPathCompression' method in 'UF_HWQUPC.java'. The code includes a comment: `/** * This implements the single-pass path-halving mechanism of path compression */` and a private method `private void doPathCompression(int i) { // FIXME update parent to value of grandparent updateParent(i, getParent(getParent(i))); // END }`.
- Run Console:** Displays the results of the unit tests. It shows a list of tests that passed, including `testIsConnected01`, `testIsConnected02`, `testIsConnected03`, `testFind0`, `testFind1`, `testFind2`, `testFind3`, `testFind4`, `testFind5`, `testToString`, `testConnect01`, `testConnect02`, and `testConnected01`. The total result is `Tests passed: 13 of 13 tests - 12 ms`.
- Terminal:** Shows the command `/Users/jessi/Library/Java/JavaVirtualMachines/adoptopenjdk-17/Contents/Home/bin/java ...` and the output `Process finished with exit code 0`.