Program Structures and Algorithms
Spring 2023(SEC 1)

NAME: Harshil Shah
NUID: 002780887
Assignment: 4

Task: Your task is

Step 1:
(a) Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF_HWQUPC. All you have to do is to fill in the sections marked with // TO BE IMPLEMENTED ... // ...END IMPLEMENTATION.

(b) Check that the unit tests for this class all work. You must show "green" test results in your submission (screenshot is OK).

Step 2:
Using your implementation of UF_HWQUPC, develop a UF ("union-find") client that takes an integer value n from the command line to determine the number of "sites." Then generates random pairs of integers between 0 and n-1, calling connected() to determine if they are connected and union() if not. Loop until all sites are connected then print the number of connections generated. Package your program as a static method count() that takes n as the argument and returns the number of connections; and a main() that takes n from the command line, calls count() and prints the returned value. If you prefer, you can create a main program that doesn't require any input and runs the experiment for a fixed set of n values. Show evidence of your run(s).

Step 3:
Determine the relationship between the number of objects (n) and the number of pairs (m) generated to accomplish this (i.e. to reduce the number of components from n to 1). Justify your conclusion in terms of your observations and what you think might be going on.

Relationship Conclusion:

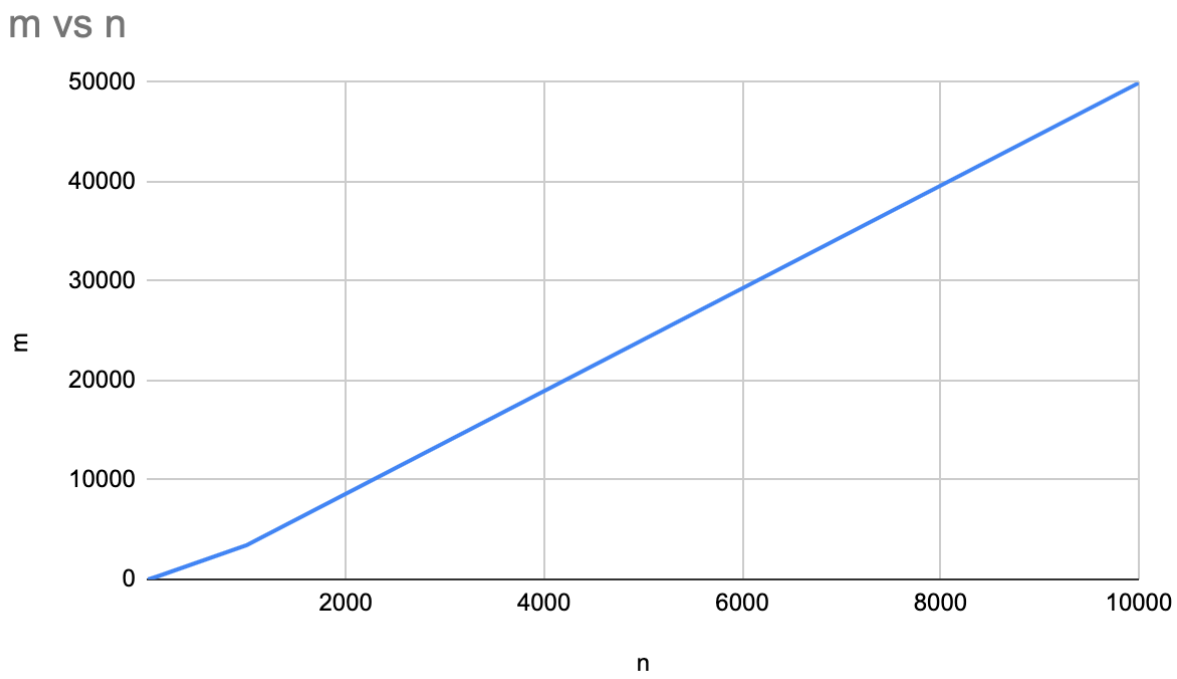
1. The number of pairs (m) generated to reduce the number of components from n to 1 is proportional to n log n, because the union operation needs to be performed log n times for each of the n objects to reduce the number of components to 1.
2. However, the actual number of pairs generated may be less than n log n in practice, because the height-weighted Quick Union Find with path compression algorithm uses path compression to reduce the height of the trees. This results in a smaller number of nodes in the trees, which in turn reduces the number of union operations required to reduce the number of components to 1.
3. Therefore, the relationship between the number of objects (n) and the number of pairs (m) generated to accomplish the reduction of components from n to 1 is proportional to n log n, but the actual number of pairs generated may be less due to the use of path compression.

Evidence to support that conclusion:

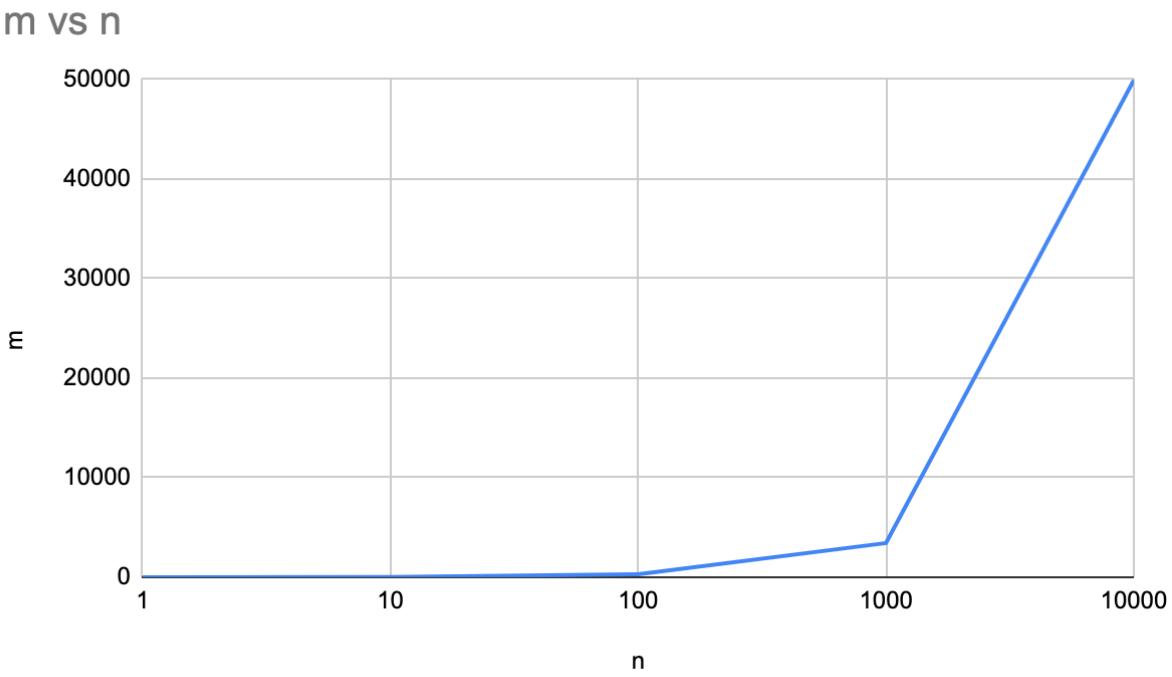
n	m
1	0
10	11
100	290
1000	3430
10000	49935

Observation: These observations show that the height-weighted Quick Union Find with path compression algorithm is efficient for reducing the number of components from n to 1, but the time complexity increases as the number of objects grows. The use of path compression helps to reduce the time complexity, but the overall time complexity still grows logarithmically with the number of objects.

Graphical Representation:

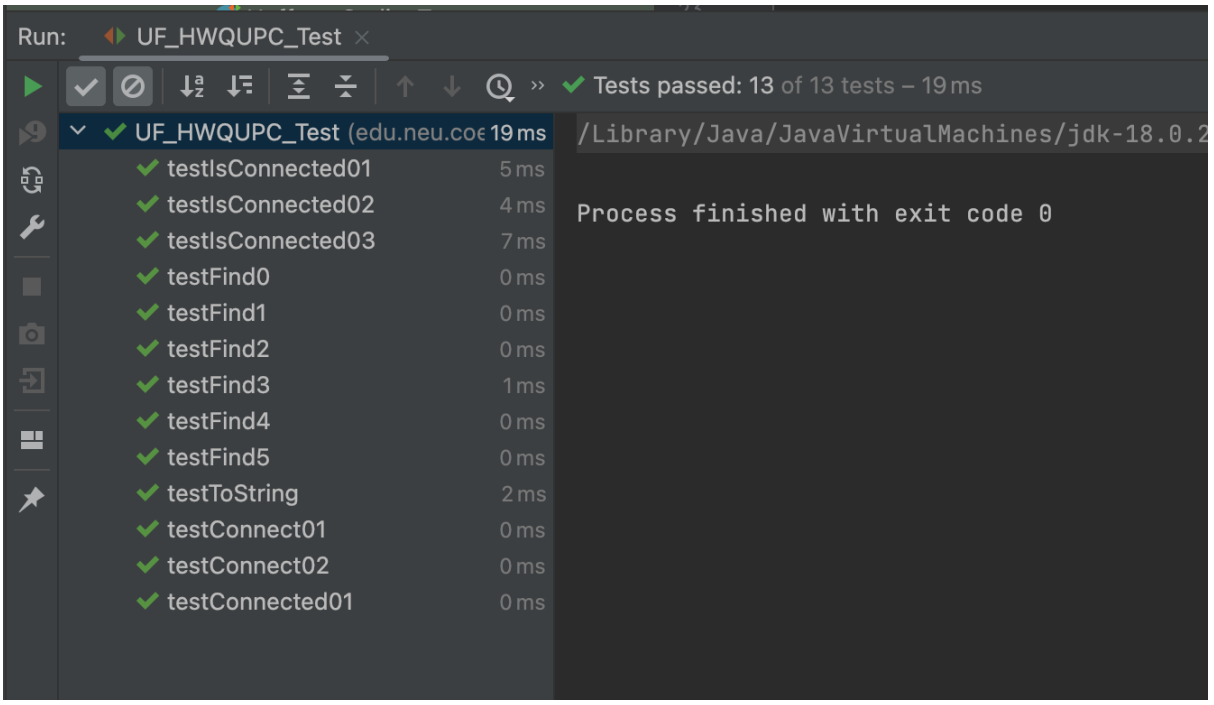


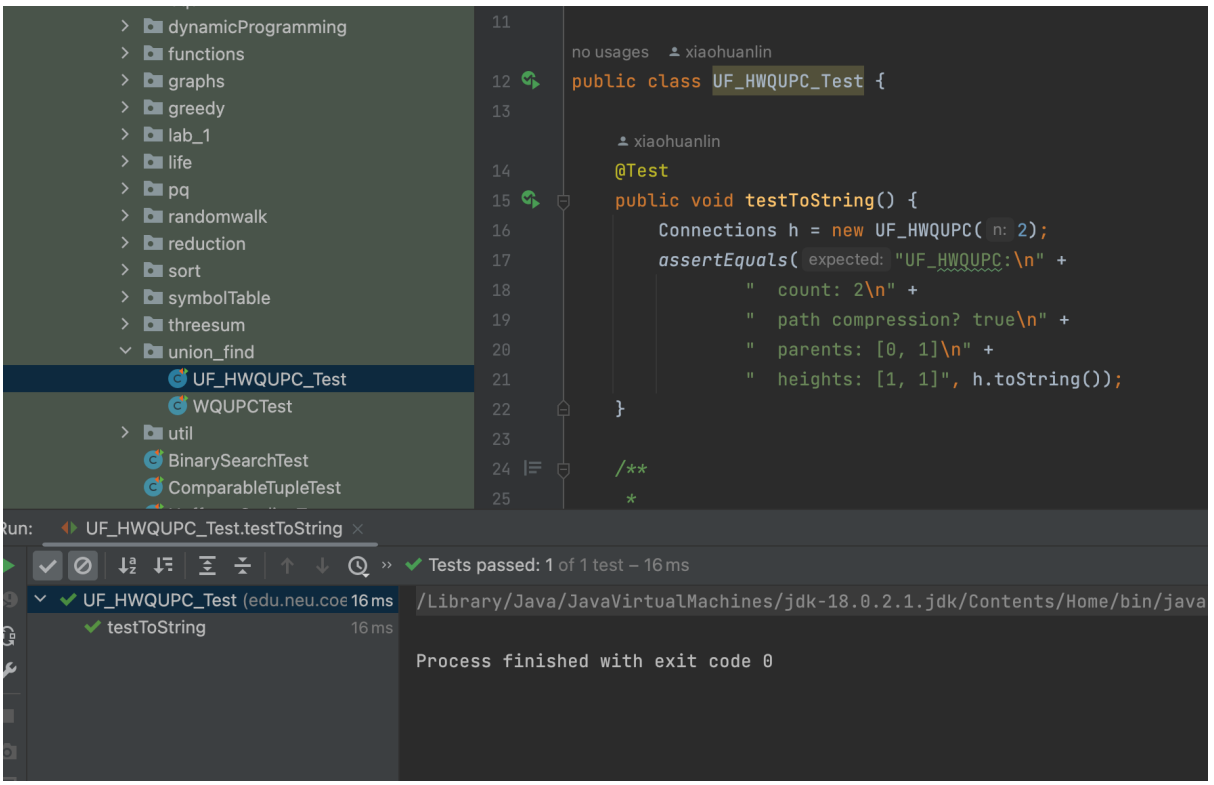
With Logarithmic Scale

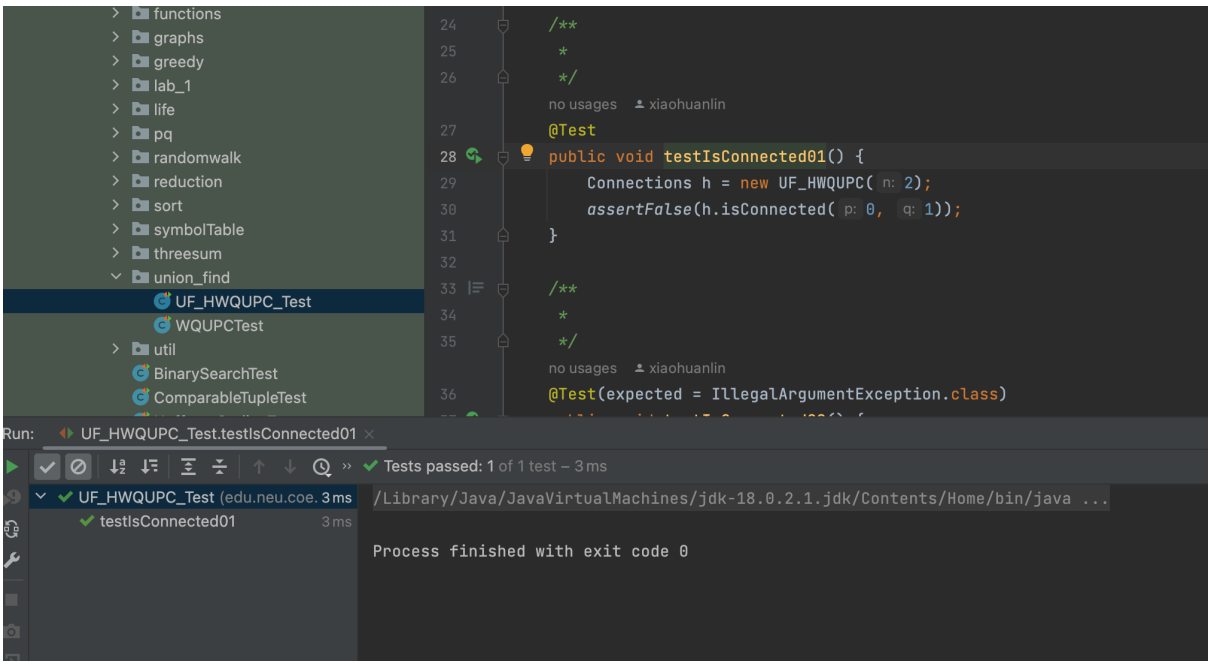


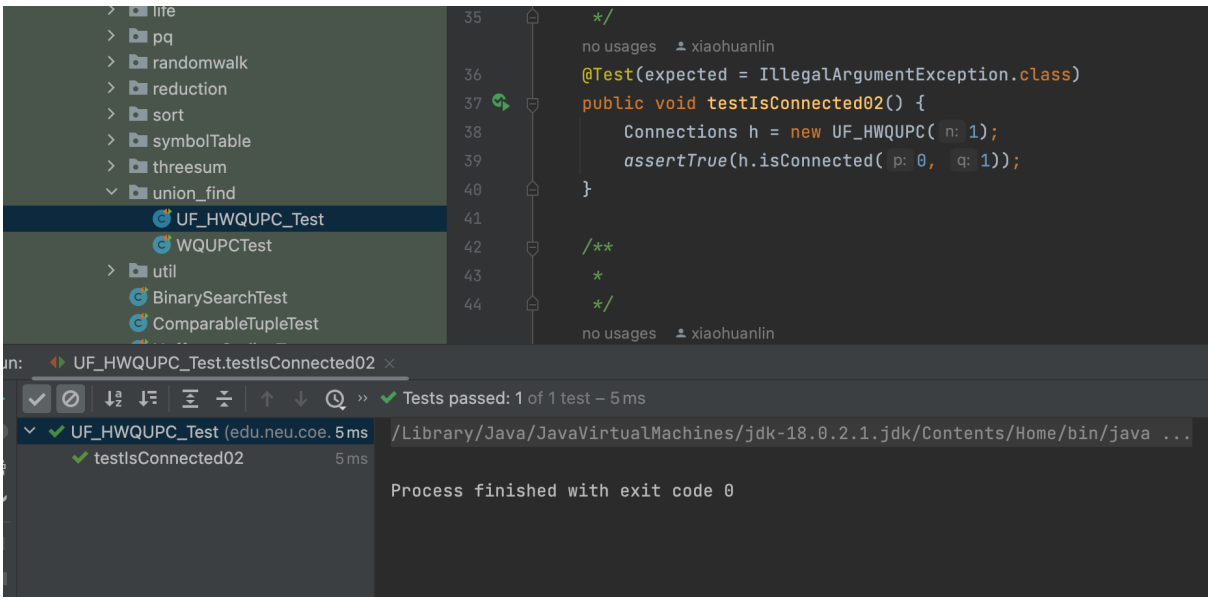
Unit Test Screenshots:

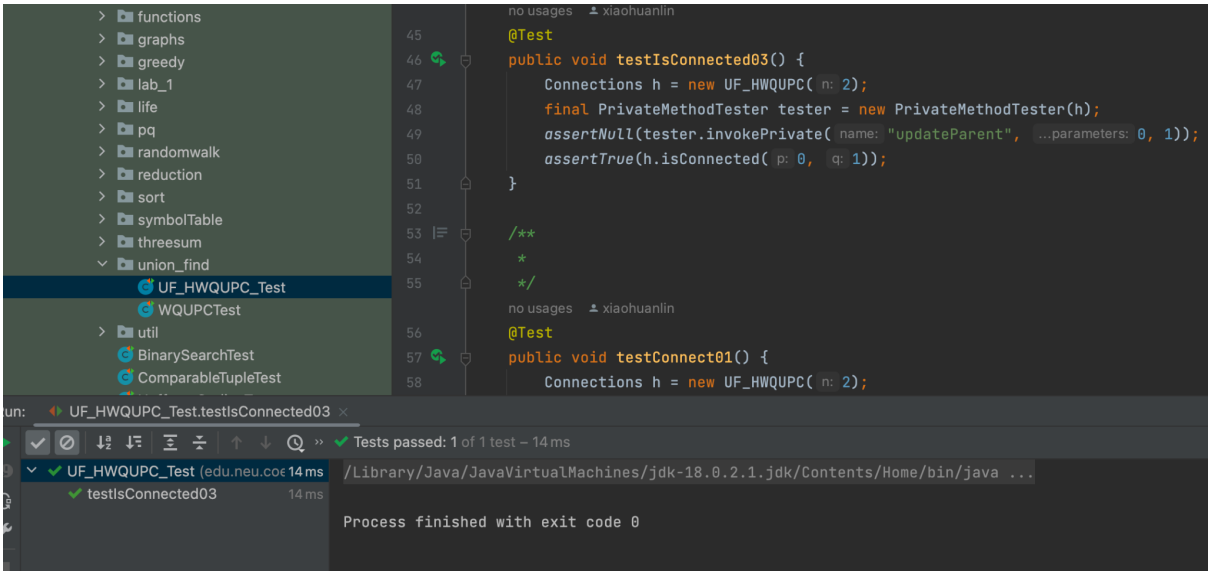
1. **Unit test cases for UF_HWQUPC**

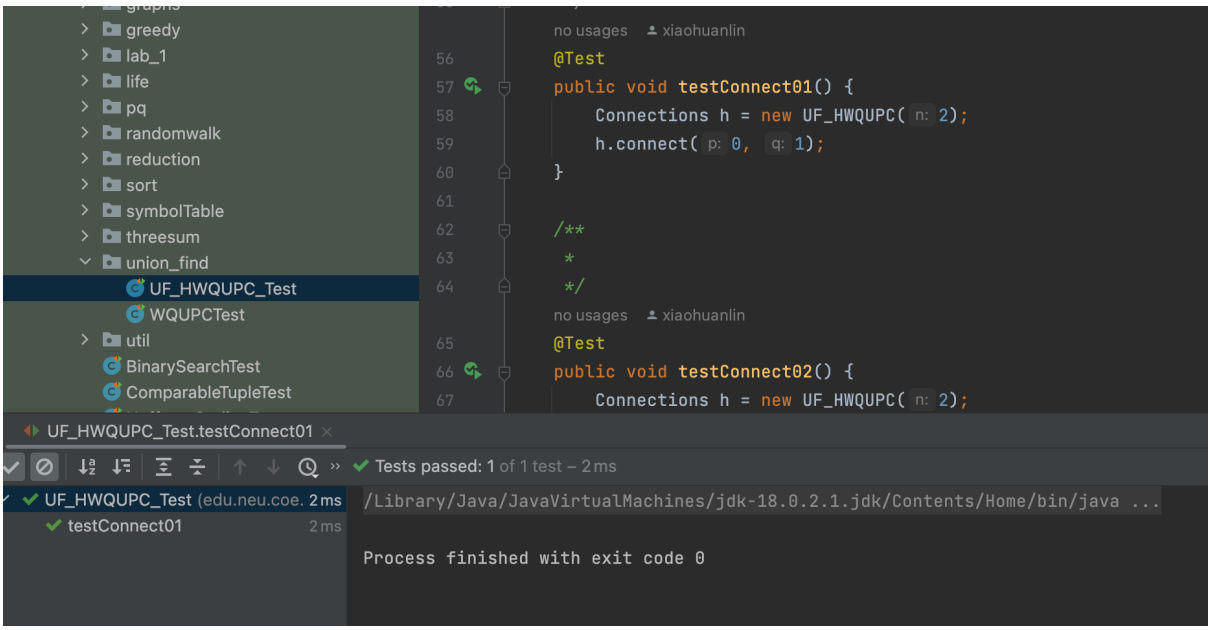


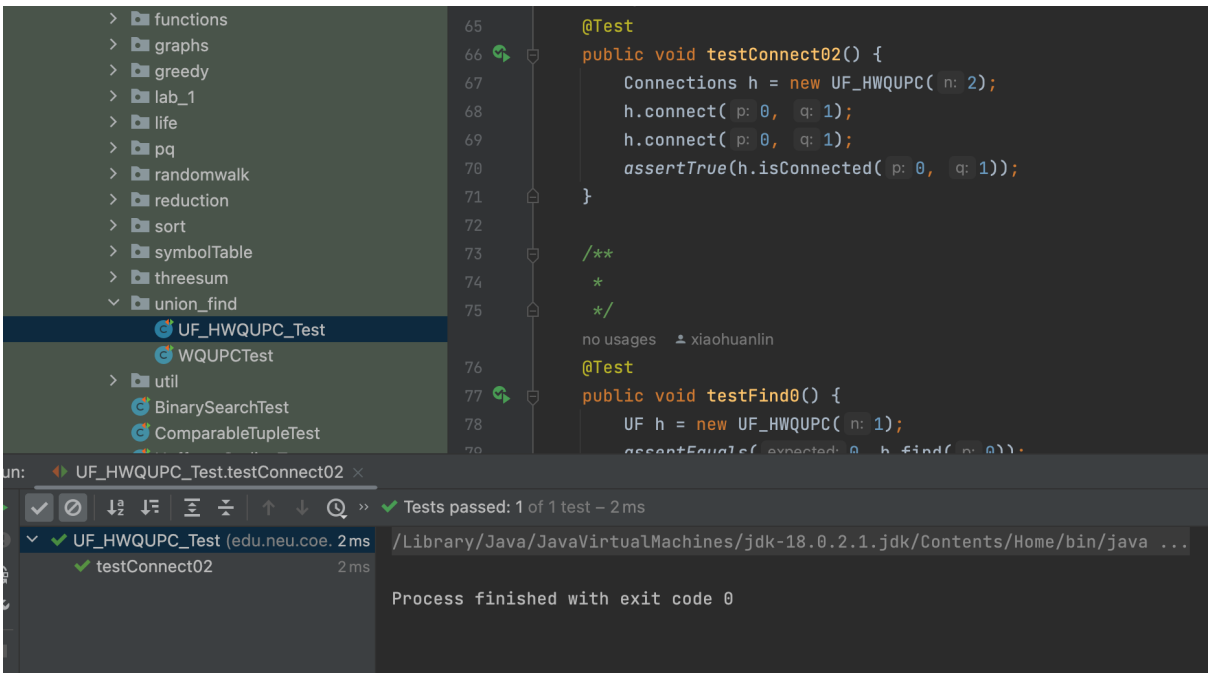

dynamicProgramming
functions
graphs
greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
no usages xiaohuanlin
public class UF_HWQUPC_Test {
xiaohuanlin
@Test
public void testToString() {
Connections h = new UF_HWQUPC(n: 2);
assertEquals(expected: "UF_HWQUPC:\n" +
" count: 2\n" +
" path compression? true\n" +
" parents: [0, 1]\n" +
" heights: [1, 1]", h.toString());
}
/**
*
Run: UF_HWQUPC_Test.testToString
Tests passed: 1 of 1 test – 16 ms
UF_HWQUPC_Test (edu.neu.coe 16 ms
testToString 16 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java
Process finished with exit code 0

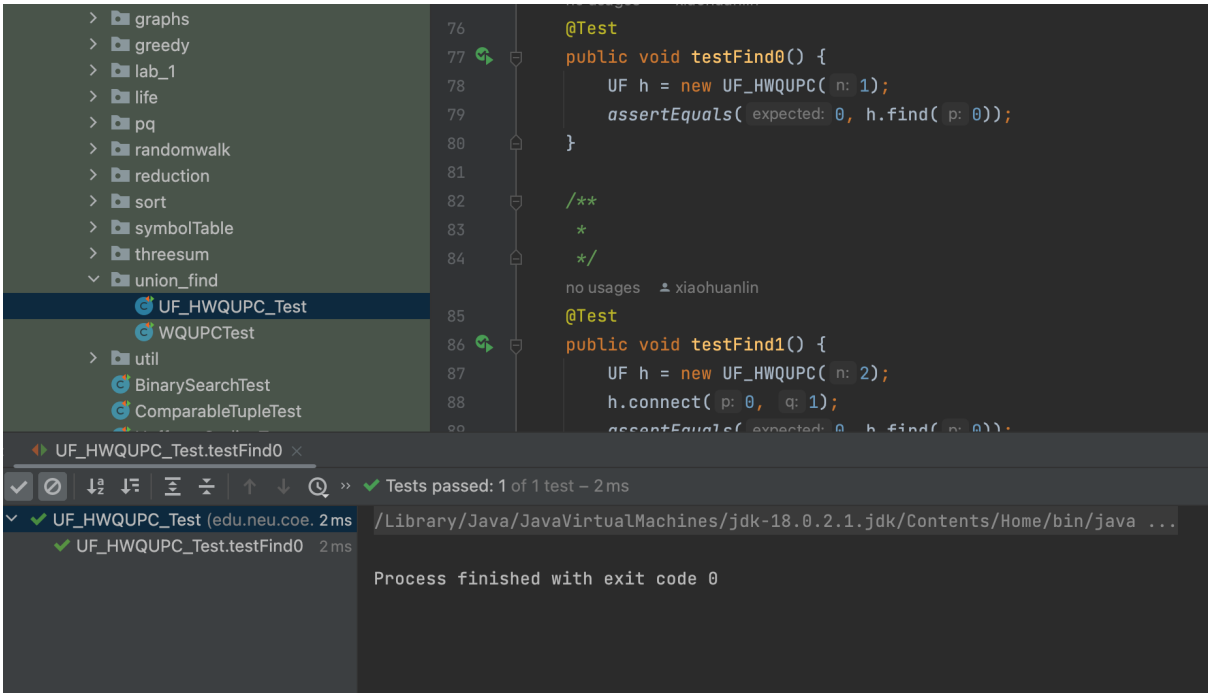

functions
graphs
greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
/**
*
*/
no usages xiaohuanlin
@Test
public void testIsConnected01() {
Connections h = new UF_HWQUPC(n: 2);
assertFalse(h.isConnected(p: 0, q: 1));
}
/**
*
*/
no usages xiaohuanlin
@Test(expected = IllegalArgumentException.class)
Run: UF_HWQUPC_Test.testIsConnected01
Tests passed: 1 of 1 test – 3 ms
UF_HWQUPC_Test (edu.neu.coe. 3 ms
testIsConnected01 3 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0

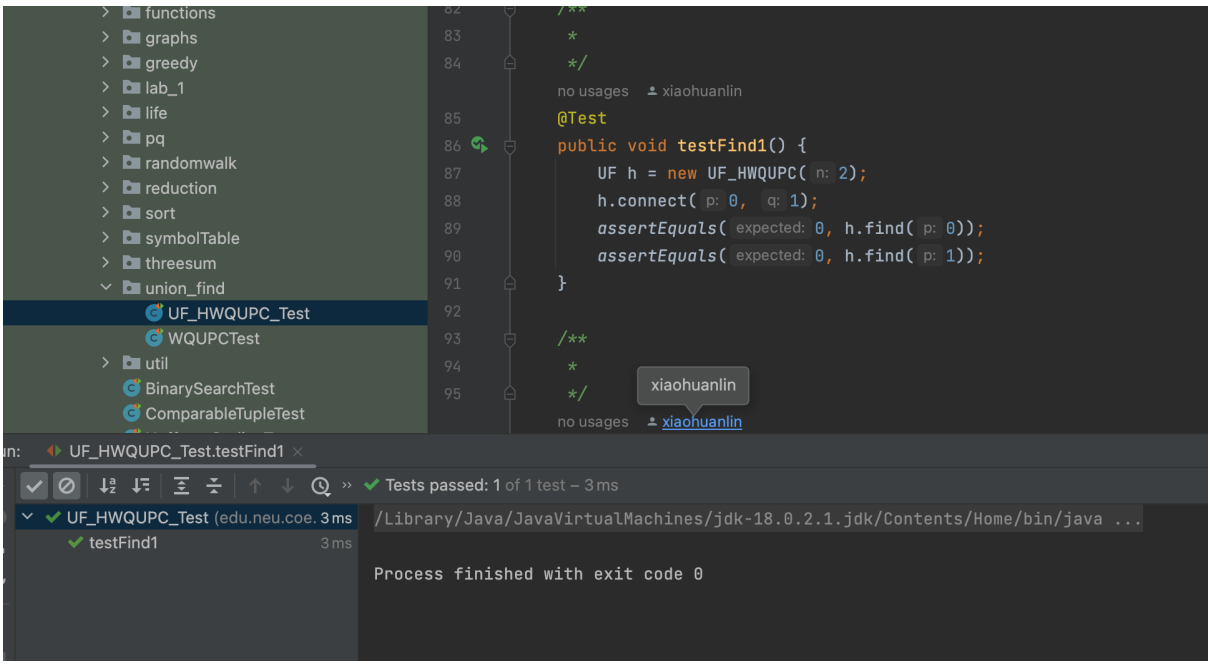

life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
*/
no usages xiaohuanlin
@Test(expected = IllegalArgumentException.class)
public void testIsConnected02() {
Connections h = new UF_HWQUPC(n: 1);
assertTrue(h.isConnected(p: 0, q: 1));
}
/**
*
*/
no usages xiaohuanlin
UF_HWQUPC_Test.testIsConnected02
Tests passed: 1 of 1 test – 5 ms
UF_HWQUPC_Test (edu.neu.coe. 5 ms
testIsConnected02 5 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0

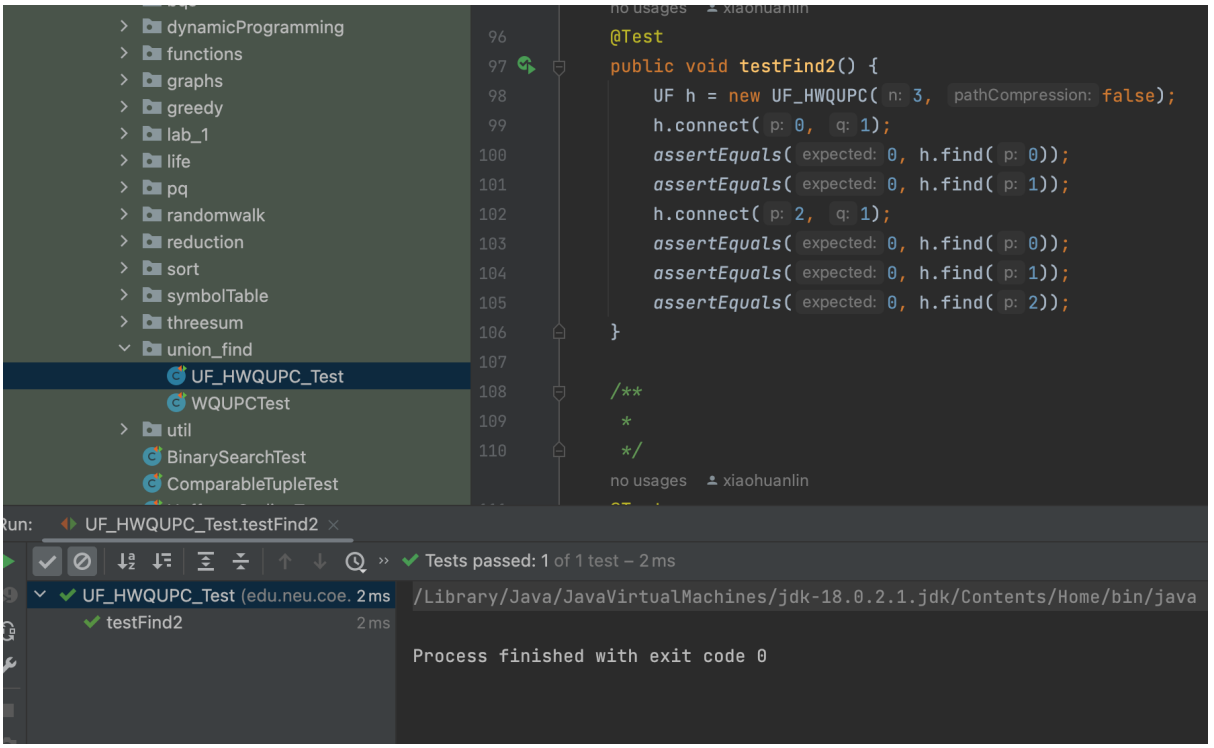

functions
graphs
greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
no usages xiaohuanlin
@Test
public void testIsConnected03() {
Connections h = new UF_HWQUPC(n: 2);
final PrivateMethodTester tester = new PrivateMethodTester(h);
assertNull(tester.invokePrivate(name: "updateParent", ...parameters: 0, 1));
assertTrue(h.isConnected(p: 0, q: 1));
}
/**
*
*/
no usages xiaohuanlin
@Test
public void testConnect01() {
Connections h = new UF_HWQUPC(n: 2);
UF_HWQUPC_Test.testIsConnected03
Tests passed: 1 of 1 test – 14 ms
UF_HWQUPC_Test (edu.neu.coe 14 ms
testIsConnected03 14 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0

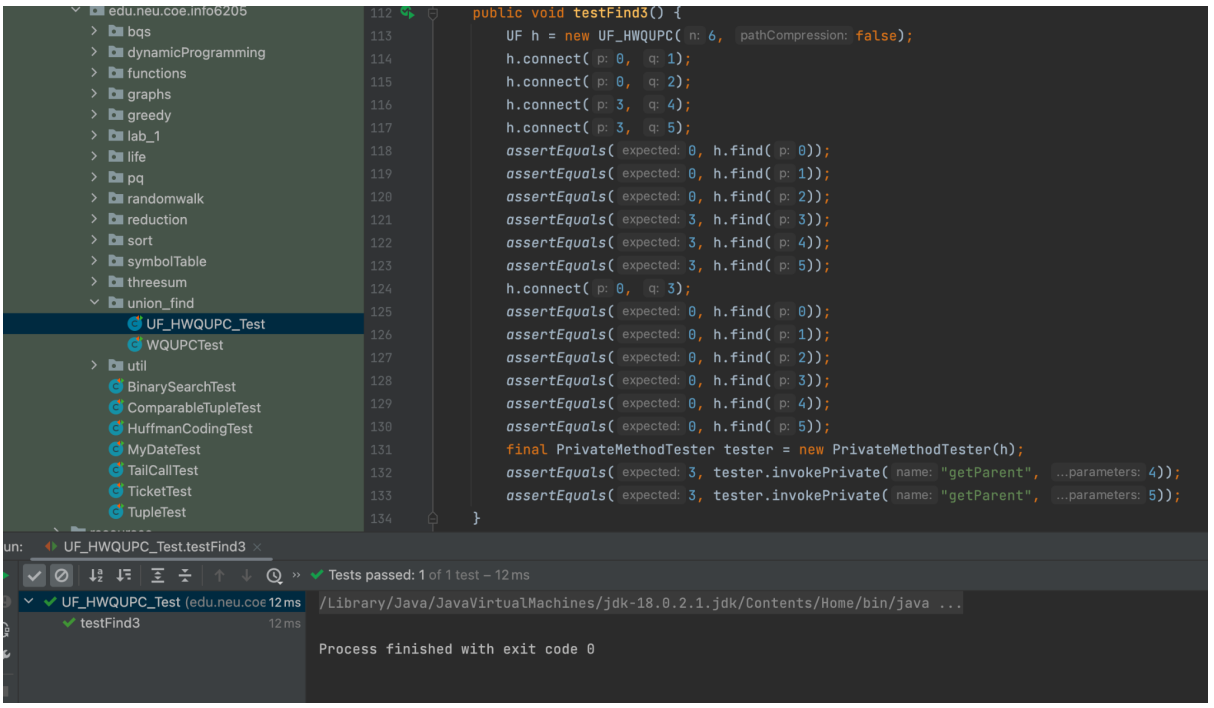

greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
no usages xiaohuanlin
@Test
public void testConnect01() {
Connections h = new UF_HWQUPC(n: 2);
h.connect(p: 0, q: 1);
}
/**
*
*/
no usages xiaohuanlin
@Test
public void testConnect02() {
Connections h = new UF_HWQUPC(n: 2);
UF_HWQUPC_Test.testConnect01
Tests passed: 1 of 1 test – 2 ms
UF_HWQUPC_Test (edu.neu.coe. 2 ms
testConnect01 2 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0

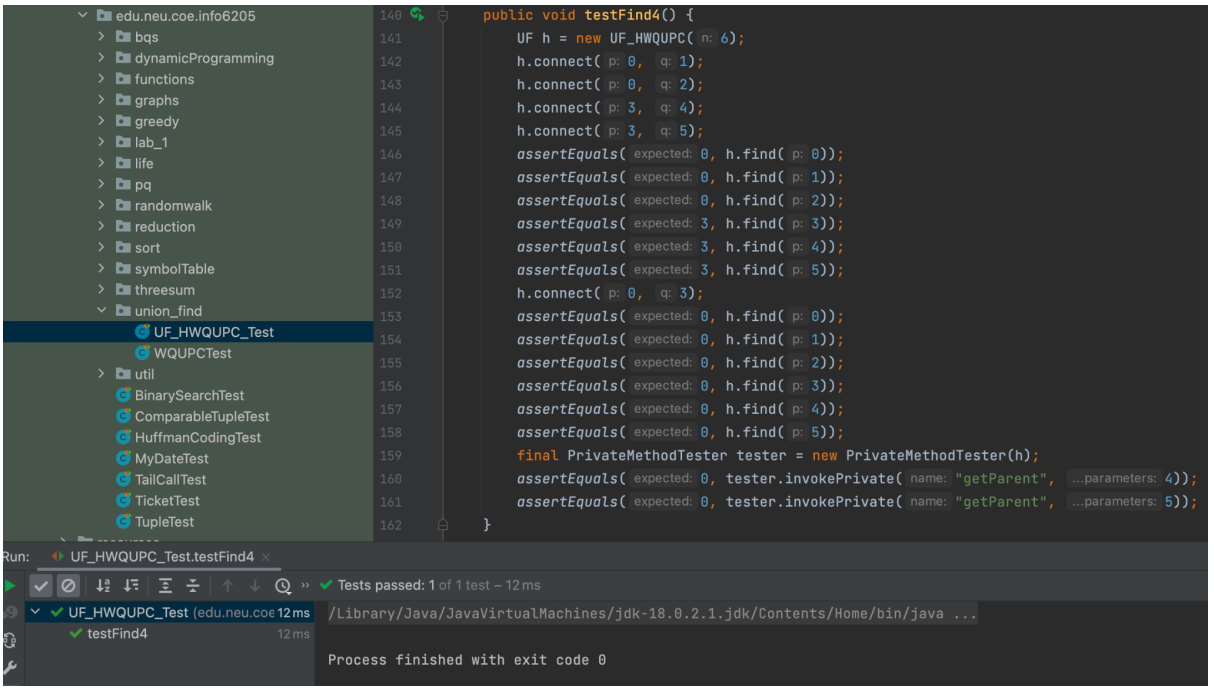

functions
graphs
greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
@Test
public void testConnect02() {
Connections h = new UF_HWQUPC(n: 2);
h.connect(p: 0, q: 1);
h.connect(p: 0, q: 1);
assertTrue(h.isConnected(p: 0, q: 1));
}
/**
*
*/
no usages xiaohuanlin
@Test
public void testFind0() {
UF h = new UF_HWQUPC(n: 1);
UF_HWQUPC_Test.testConnect02
Tests passed: 1 of 1 test – 2 ms
UF_HWQUPC_Test (edu.neu.coe. 2 ms
testConnect02 2 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0

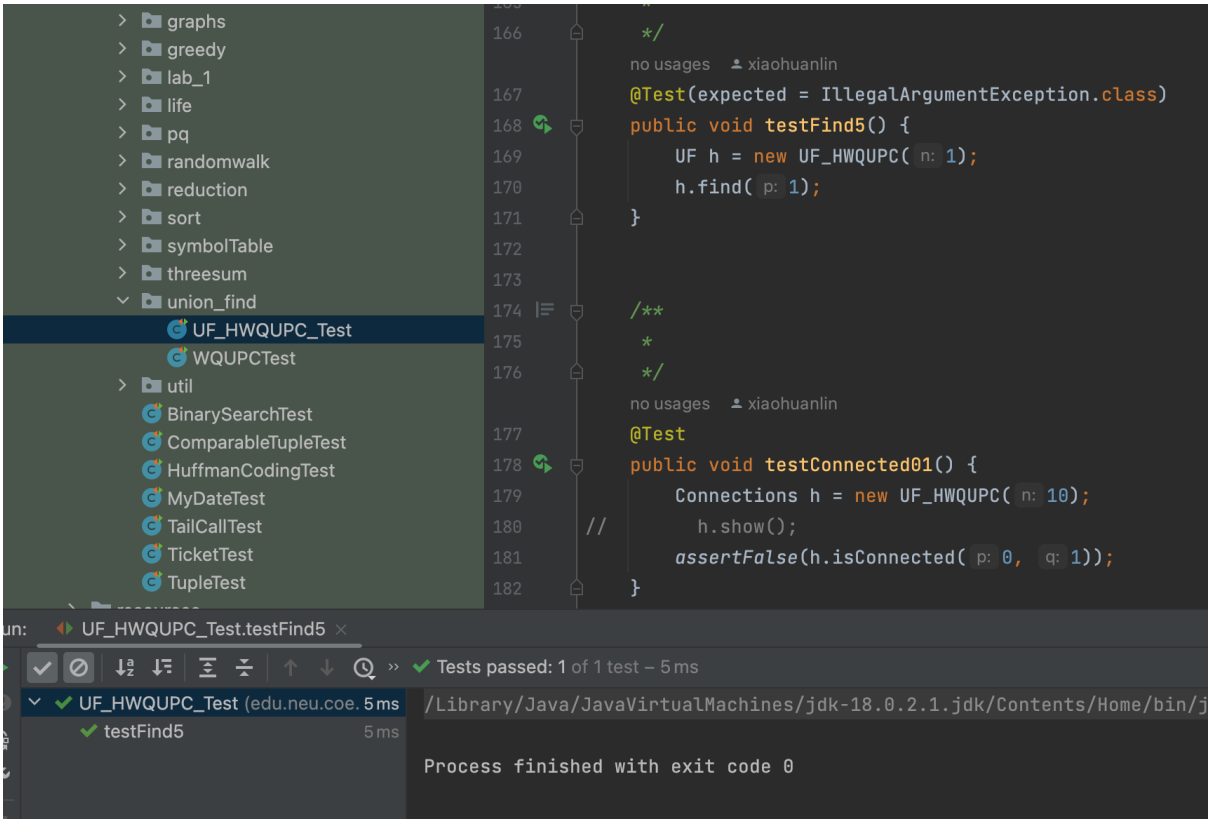

graphs
greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
@Test
public void testFind0() {
UF h = new UF_HWQUPC(n: 1);
assertEquals(expected: 0, h.find(p: 0));
}
/**
*
*/
no usages xiaohuanlin
@Test
public void testFind1() {
UF h = new UF_HWQUPC(n: 2);
h.connect(p: 0, q: 1);
UF_HWQUPC_Test.testFind0
Tests passed: 1 of 1 test – 2 ms
UF_HWQUPC_Test (edu.neu.coe. 2 ms
UF_HWQUPC_Test.testFind0 2 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0

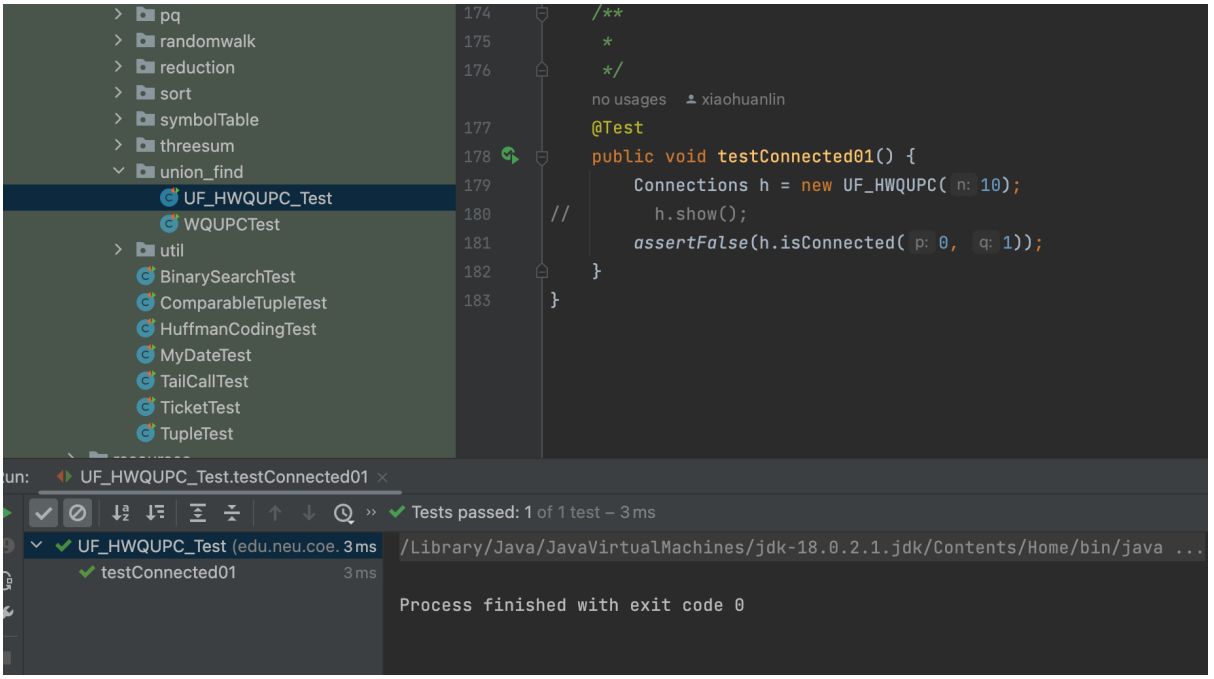

functions
graphs
greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
/**
*
*/
no usages xiaohuanlin
@Test
public void testFind1() {
UF h = new UF_HWQUPC(n: 2);
h.connect(p: 0, q: 1);
assertEquals(expected: 0, h.find(p: 0));
assertEquals(expected: 0, h.find(p: 1));
}
/**
*
*/
xiaohuanlin
no usages xiaohuanlin
UF_HWQUPC_Test.testFind1
Tests passed: 1 of 1 test – 3 ms
UF_HWQUPC_Test (edu.neu.coe. 3 ms
testFind1 3 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0


dynamicProgramming
functions
graphs
greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
@Test
public void testFind2() {
UF h = new UF_HWQUPC(n: 3, pathCompression: false);
h.connect(p: 0, q: 1);
assertEquals(expected: 0, h.find(p: 0));
assertEquals(expected: 0, h.find(p: 1));
h.connect(p: 2, q: 1);
assertEquals(expected: 0, h.find(p: 0));
assertEquals(expected: 0, h.find(p: 1));
assertEquals(expected: 0, h.find(p: 2));
}
/**
*
*/
no usages xiaohuanlin
Run: UF_HWQUPC_Test.testFind2
Tests passed: 1 of 1 test – 2 ms
UF_HWQUPC_Test (edu.neu.coe. 2 ms
testFind2 2 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java
Process finished with exit code 0

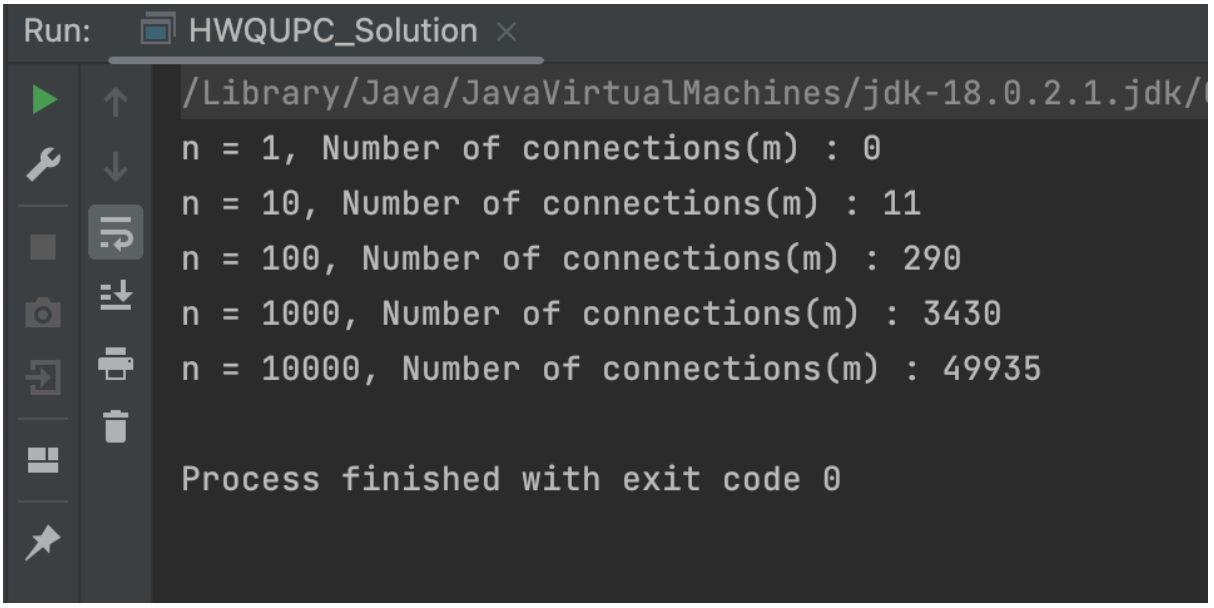

edu.neu.coe.info6205
bqs
dynamicProgramming
functions
graphs
greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
public void testFind3() {
UF h = new UF_HWQUPC(n: 6, pathCompression: false);
h.connect(p: 0, q: 1);
h.connect(p: 0, q: 2);
h.connect(p: 3, q: 4);
h.connect(p: 3, q: 5);
assertEquals(expected: 0, h.find(p: 0));
assertEquals(expected: 0, h.find(p: 1));
assertEquals(expected: 0, h.find(p: 2));
assertEquals(expected: 3, h.find(p: 3));
assertEquals(expected: 3, h.find(p: 4));
assertEquals(expected: 3, h.find(p: 5));
h.connect(p: 0, q: 3);
assertEquals(expected: 0, h.find(p: 0));
assertEquals(expected: 0, h.find(p: 1));
assertEquals(expected: 0, h.find(p: 2));
assertEquals(expected: 0, h.find(p: 3));
assertEquals(expected: 0, h.find(p: 4));
assertEquals(expected: 0, h.find(p: 5));
final PrivateMethodTester tester = new PrivateMethodTester(h);
assertEquals(expected: 3, tester.invokePrivate(name: "getParent", ...parameters: 4));
assertEquals(expected: 3, tester.invokePrivate(name: "getParent", ...parameters: 5));
}
un: UF_HWQUPC_Test.testFind3
Tests passed: 1 of 1 test – 12 ms
UF_HWQUPC_Test (edu.neu.coe 12 ms
testFind3 12 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0


edu.neu.coe.info6205
bqs
dynamicProgramming
functions
graphs
greedy
lab_1
life
pq
randomwalk
reduction
sort
symbolTable
threesum
union_find
UF_HWQUPC_Test
WQUPCTest
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
public void testFind4() {
UF h = new UF_HWQUPC(n: 6);
h.connect(p: 0, q: 1);
h.connect(p: 0, q: 2);
h.connect(p: 3, q: 4);
h.connect(p: 3, q: 5);
assertEquals(expected: 0, h.find(p: 0));
assertEquals(expected: 0, h.find(p: 1));
assertEquals(expected: 0, h.find(p: 2));
assertEquals(expected: 3, h.find(p: 3));
assertEquals(expected: 3, h.find(p: 4));
assertEquals(expected: 3, h.find(p: 5));
h.connect(p: 0, q: 3);
assertEquals(expected: 0, h.find(p: 0));
assertEquals(expected: 0, h.find(p: 1));
assertEquals(expected: 0, h.find(p: 2));
assertEquals(expected: 0, h.find(p: 3));
assertEquals(expected: 0, h.find(p: 4));
assertEquals(expected: 0, h.find(p: 5));
final PrivateMethodTester tester = new PrivateMethodTester(h);
assertEquals(expected: 0, tester.invokePrivate(name: "getParent", ...parameters: 4));
assertEquals(expected: 0, tester.invokePrivate(name: "getParent", ...parameters: 5));
}
Run: UF_HWQUPC_Test.testFind4
Tests passed: 1 of 1 test – 12 ms
UF_HWQUPC_Test (edu.neu.coe 12 ms
testFind4 12 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0





UF_Client execution:


Run:
HWQUPC_Solution
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/
n = 1, Number of connections(m) : 0
n = 10, Number of connections(m) : 11
n = 100, Number of connections(m) : 290
n = 1000, Number of connections(m) : 3430
n = 10000, Number of connections(m) : 49935
Process finished with exit code 0