Program Structures and Algorithms
Spring 2023(SEC 1)

NAME: Harshil Shah
NUID: 002780887

Task: Solve 3-SUM using the Quadrithmic, Quadratic, and (bonus point) quadraticWithCalipers approaches, as shown in skeleton code in the repository.
There are also hints in the comments of the existing code. There are a number of unit tests which you should be able to run successfully.

Submit (in your own repository--see instructions elsewhere--include the source code and the unit tests of course):

(a) evidence (screenshot) of your unit tests running (try to show the actual unit test code as well as the green strip);

(b) a spreadsheet showing your timing observations--using the doubling method for at least five values of N--for each of the algorithms (include cubic); Timing should be performed either with an actual stopwatch (e.g. your iPhone) or using the Stopwatch class in the repository.

(c) your brief explanation of why the quadratic method(s) work.

Relationship Conclusion: The Quadratic methods are more efficient than the brute force method for solving the three sum problem because it takes advantage of the properties of a sorted array to reduce the number of comparisons that need to be made.

In the brute force method, you would check every possible combination of three numbers in the array to see if they add up to the target value. This would take O(n^3) time, which can be very slow for large inputs.

The Quadratic method, on the other hand, takes advantage of the fact that the input array is sorted. By iterating through the array and using two pointers to check pairs of numbers that sum to the target value, we are able to eliminate many unnecessary comparisons.

Also, since we are iterating through the array only once and for each element, we are using two pointers that move towards each other, we are reducing the number of comparisons and thus reducing the time complexity to O(n^2).

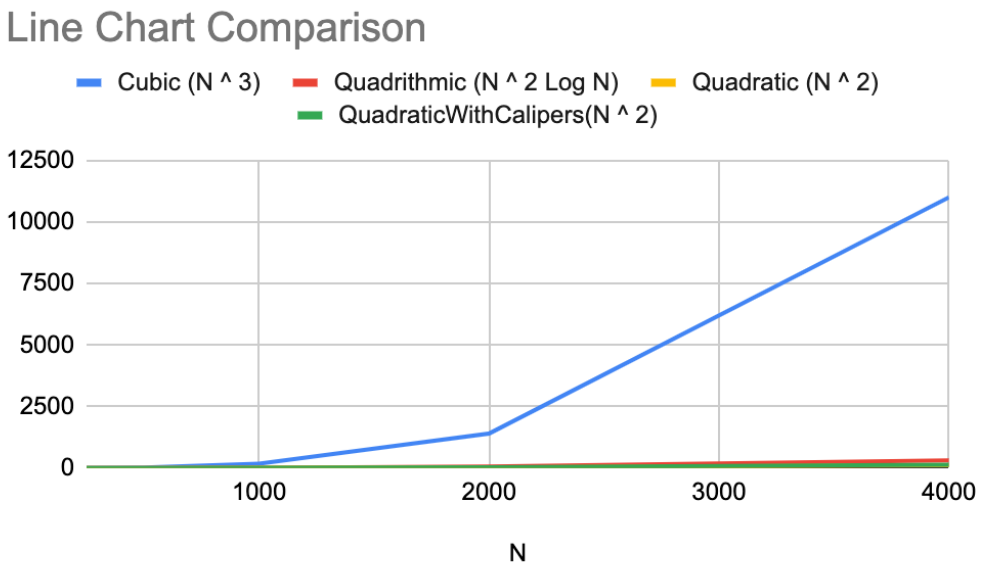
Additionally, the two-pointer method allows us to quickly discard any pairs of numbers that are not a valid solution, as soon as we find that their sum is either greater or less than the target value, this is why it's more efficient than the cubic method.

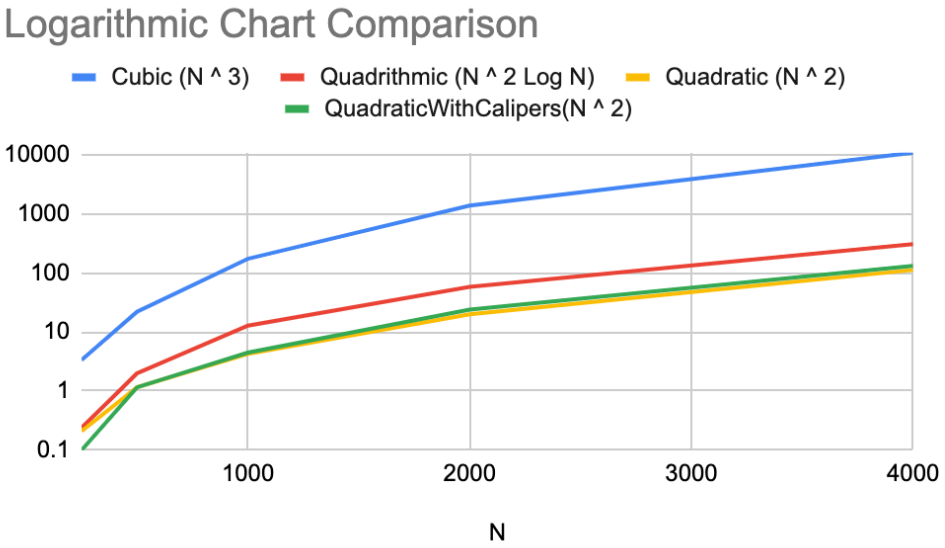
Evidence to support that conclusion:

N	Cubic (N ^ 3)	Quadrithmic (N ^ 2 Log N)	Quadratic (N ^ 2)	QuadraticWithCalipers(N ^ 2)
250	3.36	0.24	0.21	0.1
500	22.08	2	1.16	1.16
1000	174.4	12.9	4.3	4.5
2000	1397.6	58.4	20	24.1
4000	11005.6	311	113.2	132.2

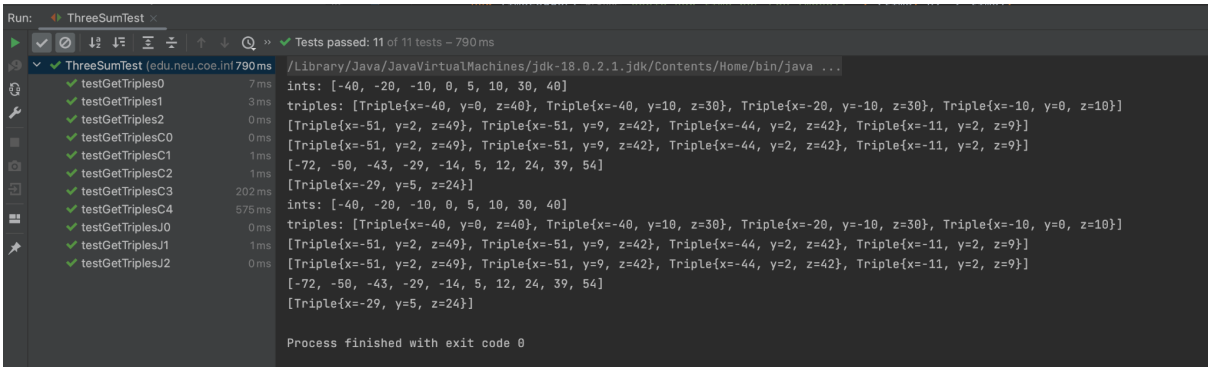
Observation: The cubic method grows at a faster rate, O(n^3), when doubling the input size, as it uses a brute force approach that calculates every combination. On the other hand, the quadratic method grows at a slower rate comparatively. This is achieved by breaking down the solution into two parts: 1) sorting the input array, which takes O(n log n) time, and 2) using a pointer approach similar to the one used to solve the two sum problem. This results in a final complexity of O(n^2) and a constant space complexity, assuming that no hashmap solution is used. However, the space complexity may vary depending on the sorting library used by different programming languages. Thus, the Quadratic approach works better compared to the Cubic approach.

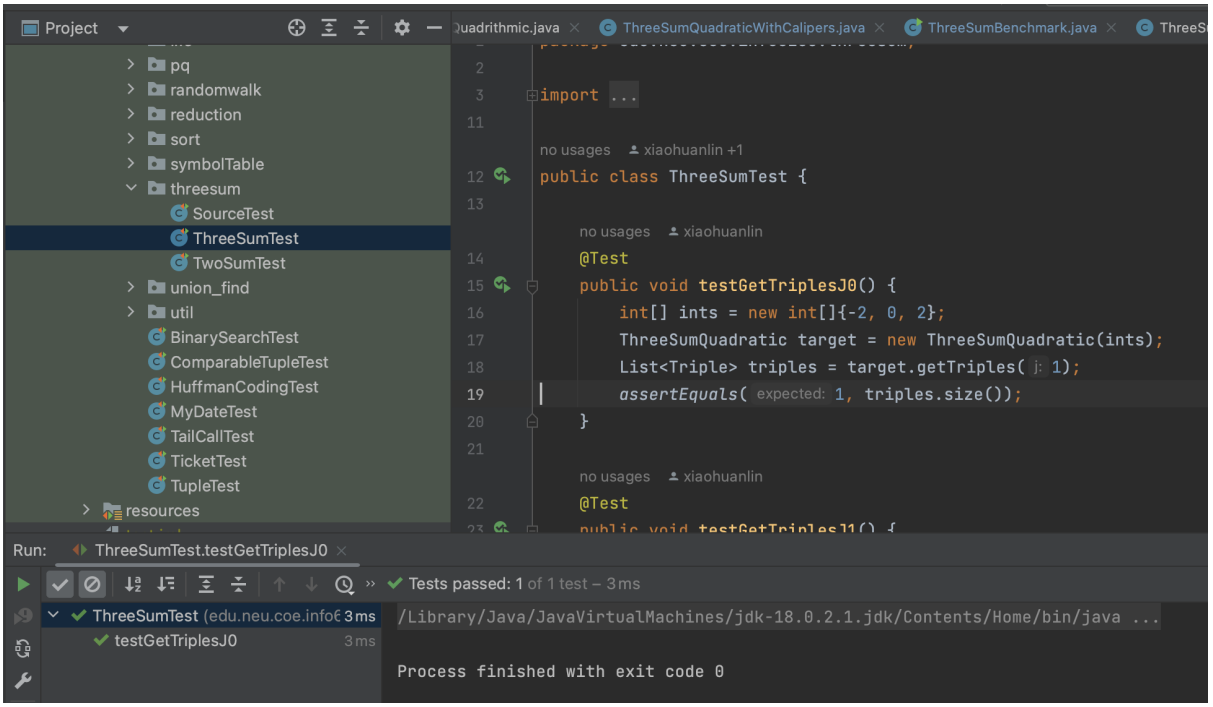
Graphical Representation:

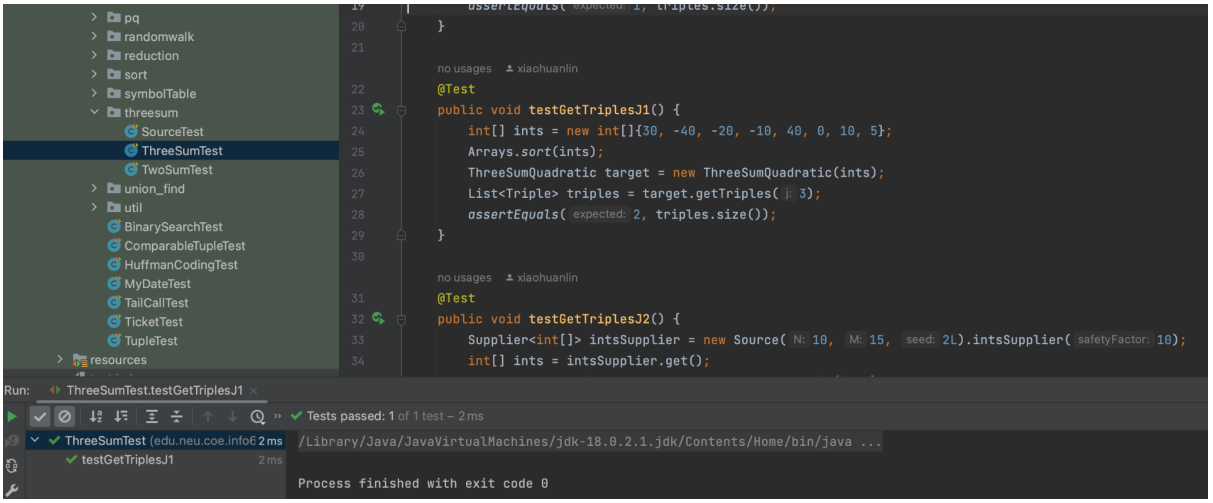


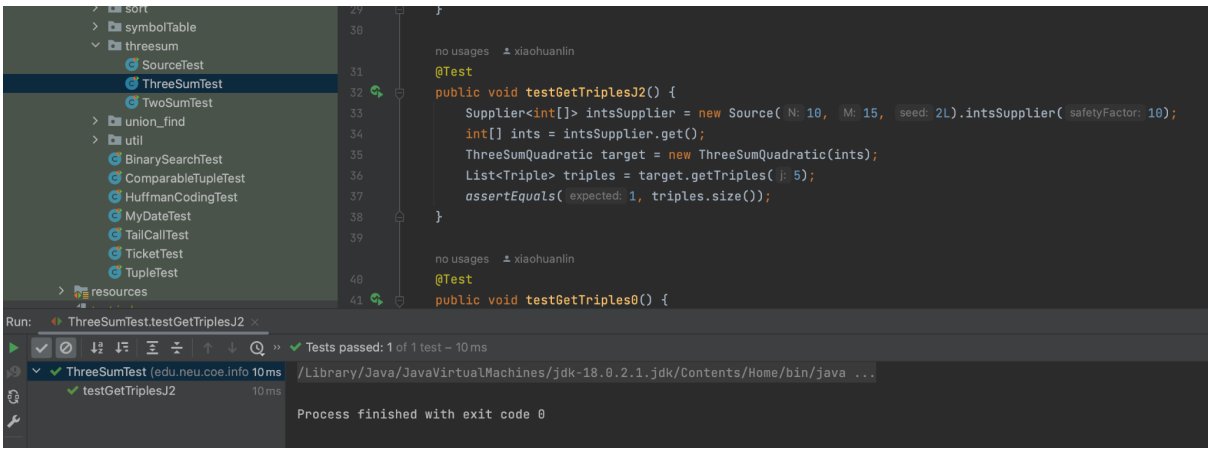


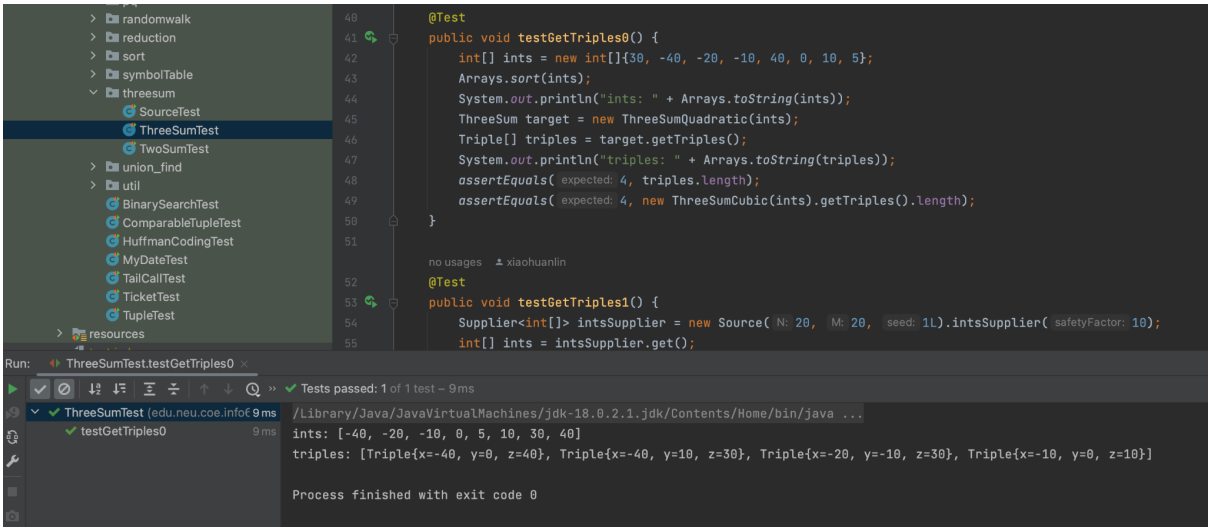
Unit Test Screenshots:

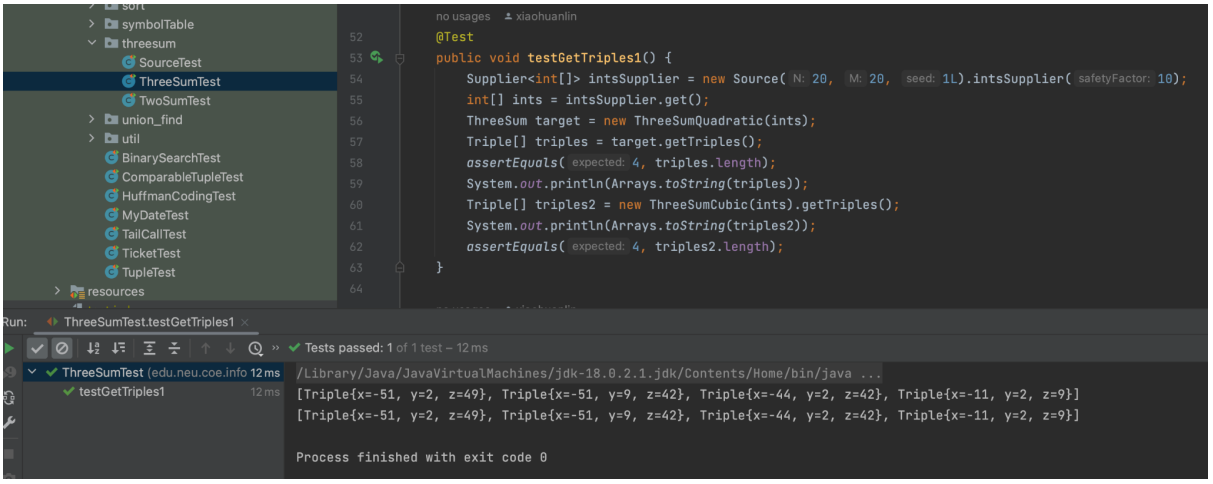


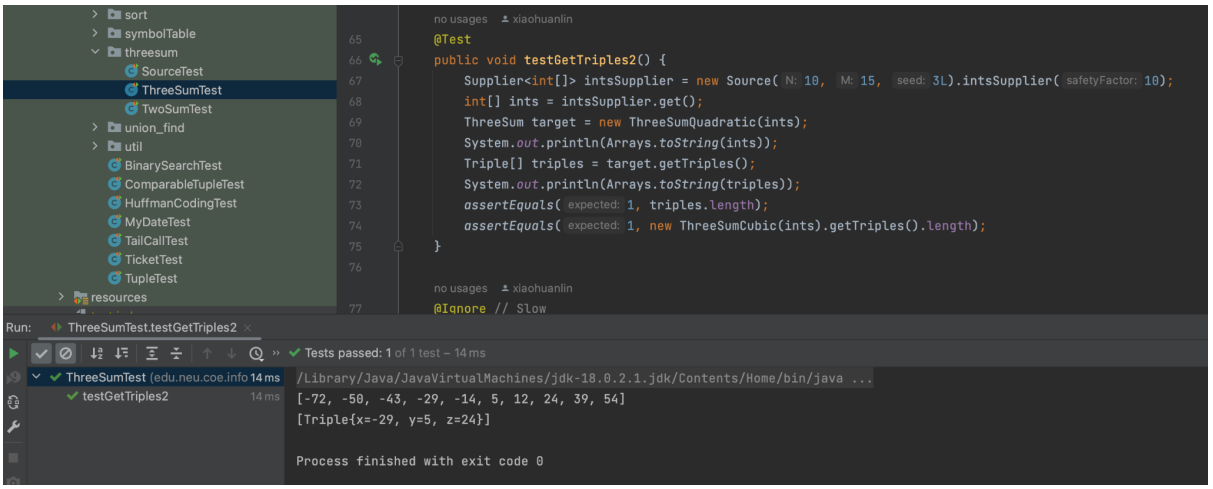

Project
pq
randomwalk
reduction
sort
symbolTable
threesum
SourceTest
ThreeSumTest
TwoSumTest
union_find
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
resources
uadrithmic.java
ThreeSumQuadraticWithCalipers.java
ThreeSumBenchmark.java
import ...
no usages xiaohuanlin +1
public class ThreeSumTest {
no usages xiaohuanlin
@Test
public void testGetTriplesJ0() {
int[] ints = new int[]{-2, 0, 2};
ThreeSumQuadratic target = new ThreeSumQuadratic(ints);
List<Triple> triples = target.getTriples(j: 1);
assertEquals(expected: 1, triples.size());
}
no usages xiaohuanlin
@Test
Run: ThreeSumTest.testGetTriplesJ0
Tests passed: 1 of 1 test – 3 ms
ThreeSumTest (edu.neu.coe.info6 3 ms
testGetTriplesJ0 3 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0

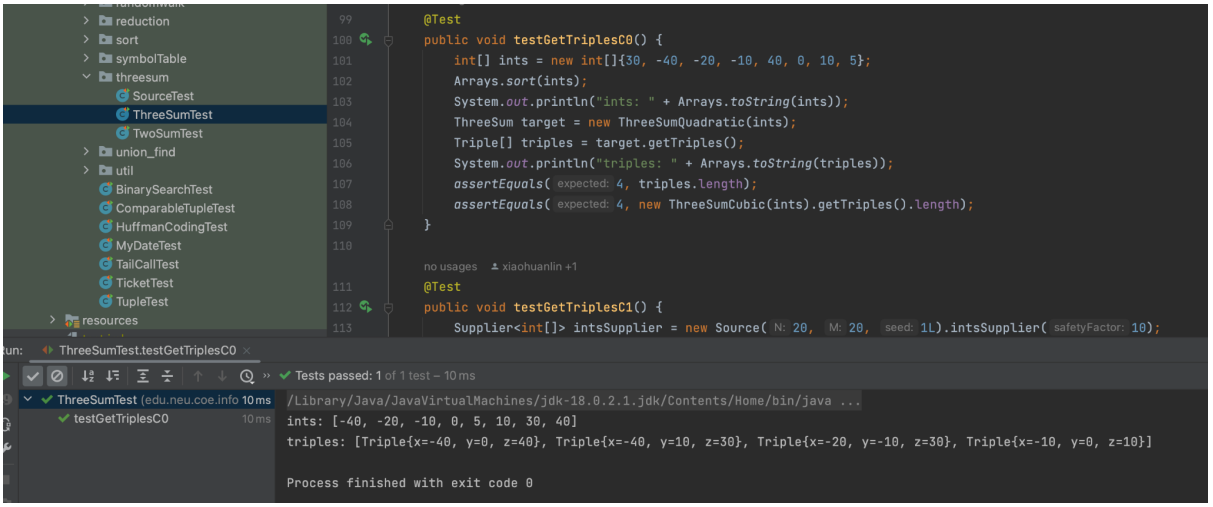

}
no usages xiaohuanlin
@Test
public void testGetTriplesJ1() {
int[] ints = new int[]{30, -40, -20, -10, 40, 0, 10, 5};
Arrays.sort(ints);
ThreeSumQuadratic target = new ThreeSumQuadratic(ints);
List<Triple> triples = target.getTriples(j: 3);
assertEquals(expected: 2, triples.size());
}
no usages xiaohuanlin
@Test
public void testGetTriplesJ2() {
Supplier<int[]> intsSupplier = new Source(N: 10, M: 15, seed: 2L).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
Run: ThreeSumTest.testGetTriplesJ1
Tests passed: 1 of 1 test – 2 ms
ThreeSumTest (edu.neu.coe.info6 2 ms
testGetTriplesJ1 2 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0

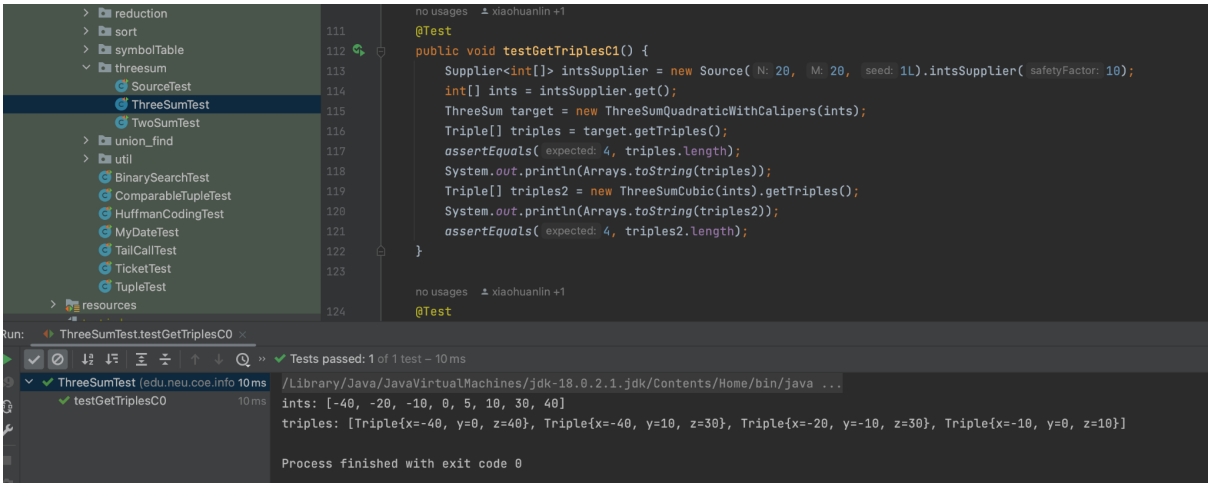

}
no usages xiaohuanlin
@Test
public void testGetTriplesJ2() {
Supplier<int[]> intsSupplier = new Source(N: 10, M: 15, seed: 2L).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
ThreeSumQuadratic target = new ThreeSumQuadratic(ints);
List<Triple> triples = target.getTriples(j: 5);
assertEquals(expected: 1, triples.size());
}
no usages xiaohuanlin
@Test
public void testGetTriples0() {
Run: ThreeSumTest.testGetTriplesJ2
Tests passed: 1 of 1 test – 10 ms
ThreeSumTest (edu.neu.coe.info 10 ms
testGetTriplesJ2 10 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0

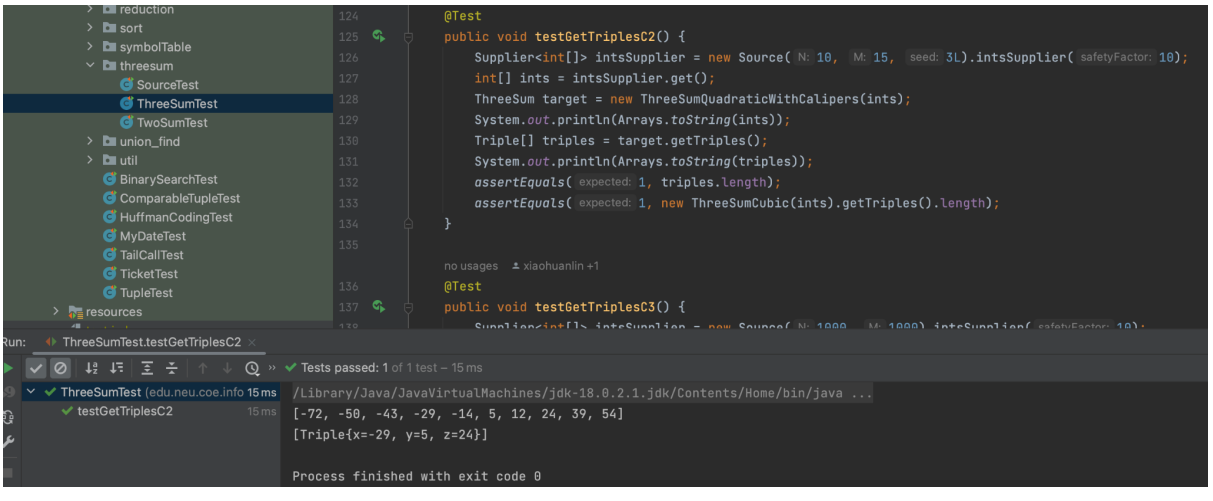

randomwalk
reduction
sort
symbolTable
threesum
SourceTest
ThreeSumTest
TwoSumTest
union_find
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
resources
@Test
public void testGetTriples0() {
int[] ints = new int[]{30, -40, -20, -10, 40, 0, 10, 5};
Arrays.sort(ints);
System.out.println("ints: " + Arrays.toString(ints));
ThreeSum target = new ThreeSumQuadratic(ints);
Triple[] triples = target.getTriples();
System.out.println("triples: " + Arrays.toString(triples));
assertEquals(expected: 4, triples.length);
assertEquals(expected: 4, new ThreeSumCubic(ints).getTriples().length);
}
no usages xiaohuanlin
@Test
public void testGetTriples1() {
Supplier<int[]> intsSupplier = new Source(N: 20, M: 20, seed: 1L).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
Run: ThreeSumTest.testGetTriples0
Tests passed: 1 of 1 test – 9 ms
ThreeSumTest (edu.neu.coe.info 9 ms
testGetTriples0 9 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
ints: [-40, -20, -10, 0, 5, 10, 30, 40]
triples: [Triple{x=-40, y=0, z=40}, Triple{x=-40, y=10, z=30}, Triple{x=-20, y=-10, z=30}, Triple{x=-10, y=0, z=10}]
Process finished with exit code 0

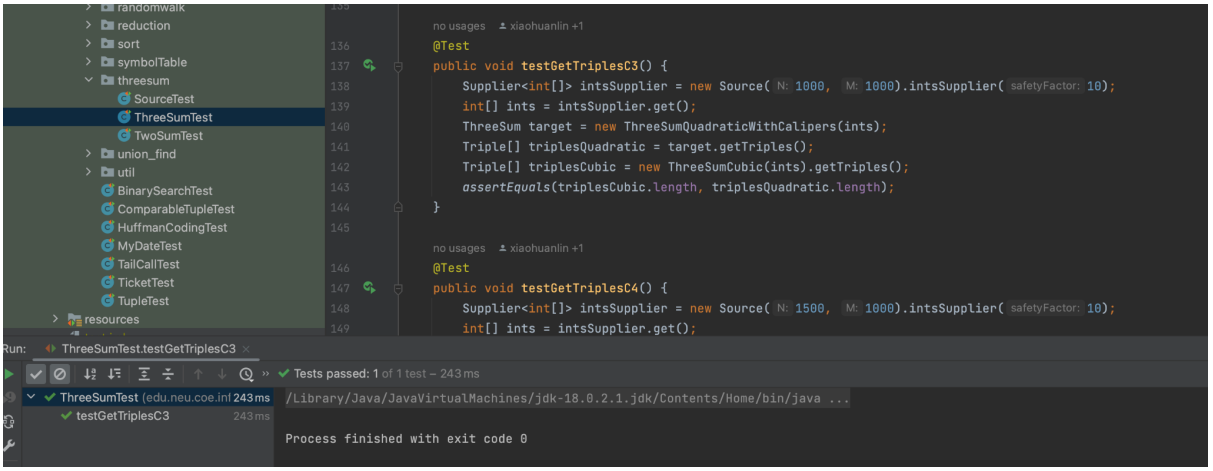

symbolTable
threesum
SourceTest
ThreeSumTest
TwoSumTest
union_find
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
resources
no usages xiaohuanlin
@Test
public void testGetTriples1() {
Supplier<int[]> intsSupplier = new Source(N: 20, M: 20, seed: 1L).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
ThreeSum target = new ThreeSumQuadratic(ints);
Triple[] triples = target.getTriples();
assertEquals(expected: 4, triples.length);
System.out.println(Arrays.toString(triples));
Triple[] triples2 = new ThreeSumCubic(ints).getTriples();
System.out.println(Arrays.toString(triples2));
assertEquals(expected: 4, triples2.length);
}
Run: ThreeSumTest.testGetTriples1
Tests passed: 1 of 1 test – 12 ms
ThreeSumTest (edu.neu.coe.info 12 ms
testGetTriples1 12 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
[Triple{x=-51, y=2, z=49}, Triple{x=-51, y=9, z=42}, Triple{x=-44, y=2, z=42}, Triple{x=-11, y=2, z=9}]
[Triple{x=-51, y=2, z=49}, Triple{x=-51, y=9, z=42}, Triple{x=-44, y=2, z=42}, Triple{x=-11, y=2, z=9}]
Process finished with exit code 0

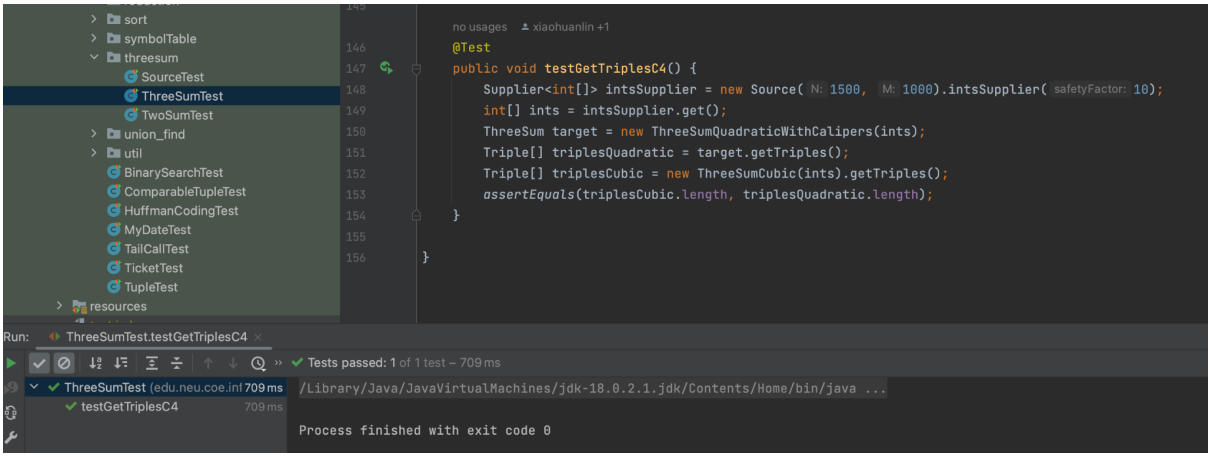

sort
symbolTable
threesum
SourceTest
ThreeSumTest
TwoSumTest
union_find
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
resources
no usages xiaohuanlin
@Test
public void testGetTriples2() {
Supplier<int[]> intsSupplier = new Source(N: 10, M: 15, seed: 3L).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
ThreeSum target = new ThreeSumQuadratic(ints);
System.out.println(Arrays.toString(ints));
Triple[] triples = target.getTriples();
System.out.println(Arrays.toString(triples));
assertEquals(expected: 1, triples.length);
assertEquals(expected: 1, new ThreeSumCubic(ints).getTriples().length);
}
no usages xiaohuanlin
@Ignore // Slow
Run: ThreeSumTest.testGetTriples2
Tests passed: 1 of 1 test – 14 ms
ThreeSumTest (edu.neu.coe.info 14 ms
testGetTriples2 14 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
[-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
[Triple{x=-29, y=5, z=24}]
Process finished with exit code 0


reduction
sort
symbolTable
threesum
SourceTest
ThreeSumTest
TwoSumTest
union_find
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
resources
@Test
public void testGetTriplesC0() {
int[] ints = new int[]{30, -40, -20, -10, 40, 0, 10, 5};
Arrays.sort(ints);
System.out.println("ints: " + Arrays.toString(ints));
ThreeSum target = new ThreeSumQuadratic(ints);
Triple[] triples = target.getTriples();
System.out.println("triples: " + Arrays.toString(triples));
assertEquals(expected: 4, triples.length);
assertEquals(expected: 4, new ThreeSumCubic(ints).getTriples().length);
}
no usages xiaohuanlin +1
@Test
public void testGetTriplesC1() {
Supplier<int[]> intsSupplier = new Source(N: 20, M: 20, seed: 1L).intsSupplier(safetyFactor: 10);
Run: ThreeSumTest.testGetTriplesC0
Tests passed: 1 of 1 test – 10 ms
ThreeSumTest (edu.neu.coe.info 10 ms
testGetTriplesC0 10 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
ints: [-40, -20, -10, 0, 5, 10, 30, 40]
triples: [Triple{x=-40, y=0, z=40}, Triple{x=-40, y=10, z=30}, Triple{x=-20, y=-10, z=30}, Triple{x=-10, y=0, z=10}]
Process finished with exit code 0


reduction
sort
symbolTable
threesum
SourceTest
ThreeSumTest
TwoSumTest
union_find
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
resources
no usages xiaohuanlin +1
@Test
public void testGetTriplesC1() {
Supplier<int[]> intsSupplier = new Source(N: 20, M: 20, seed: 1L).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
ThreeSum target = new ThreeSumQuadraticWithCalipers(ints);
Triple[] triples = target.getTriples();
assertEquals(expected: 4, triples.length);
System.out.println(Arrays.toString(triples));
Triple[] triples2 = new ThreeSumCubic(ints).getTriples();
System.out.println(Arrays.toString(triples2));
assertEquals(expected: 4, triples2.length);
}
no usages xiaohuanlin +1
@Test
Run: ThreeSumTest.testGetTriplesC0
Tests passed: 1 of 1 test – 10 ms
ThreeSumTest (edu.neu.coe.info 10 ms
testGetTriplesC0 10 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
ints: [-40, -20, -10, 0, 5, 10, 30, 40]
triples: [Triple{x=-40, y=0, z=40}, Triple{x=-40, y=10, z=30}, Triple{x=-20, y=-10, z=30}, Triple{x=-10, y=0, z=10}]
Process finished with exit code 0


reduction
sort
symbolTable
threesum
SourceTest
ThreeSumTest
TwoSumTest
union_find
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
resources
@Test
public void testGetTriplesC2() {
Supplier<int[]> intsSupplier = new Source(N: 10, M: 15, seed: 3L).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
ThreeSum target = new ThreeSumQuadraticWithCalipers(ints);
System.out.println(Arrays.toString(ints));
Triple[] triples = target.getTriples();
System.out.println(Arrays.toString(triples));
assertEquals(expected: 1, triples.length);
assertEquals(expected: 1, new ThreeSumCubic(ints).getTriples().length);
}
no usages xiaohuanlin +1
@Test
public void testGetTriplesC3() {
Run: ThreeSumTest.testGetTriplesC2
Tests passed: 1 of 1 test – 15 ms
ThreeSumTest (edu.neu.coe.info 15 ms
testGetTriplesC2 15 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
[-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
[Triple{x=-29, y=5, z=24}]
Process finished with exit code 0


randomwalk
reduction
sort
symbolTable
threesum
SourceTest
ThreeSumTest
TwoSumTest
union_find
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
resources
no usages xiaohuanlin +1
@Test
public void testGetTriplesC3() {
Supplier<int[]> intsSupplier = new Source(N: 1000, M: 1000).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
ThreeSum target = new ThreeSumQuadraticWithCalipers(ints);
Triple[] triplesQuadratic = target.getTriples();
Triple[] triplesCubic = new ThreeSumCubic(ints).getTriples();
assertEquals(triplesCubic.length, triplesQuadratic.length);
}
no usages xiaohuanlin +1
@Test
public void testGetTriplesC4() {
Supplier<int[]> intsSupplier = new Source(N: 1500, M: 1000).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
Run: ThreeSumTest.testGetTriplesC3
Tests passed: 1 of 1 test – 243 ms
ThreeSumTest (edu.neu.coe.inf 243 ms
testGetTriplesC3 243 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0


sort
symbolTable
threesum
SourceTest
ThreeSumTest
TwoSumTest
union_find
util
BinarySearchTest
ComparableTupleTest
HuffmanCodingTest
MyDateTest
TailCallTest
TicketTest
TupleTest
resources
no usages xiaohuanlin +1
@Test
public void testGetTriplesC4() {
Supplier<int[]> intsSupplier = new Source(N: 1500, M: 1000).intsSupplier(safetyFactor: 10);
int[] ints = intsSupplier.get();
ThreeSum target = new ThreeSumQuadraticWithCalipers(ints);
Triple[] triplesQuadratic = target.getTriples();
Triple[] triplesCubic = new ThreeSumCubic(ints).getTriples();
assertEquals(triplesCubic.length, triplesQuadratic.length);
}
}
Run: ThreeSumTest.testGetTriplesC4
Tests passed: 1 of 1 test – 709 ms
ThreeSumTest (edu.neu.coe.inf 709 ms
testGetTriplesC4 709 ms
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Process finished with exit code 0