

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

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 $O(n^2)$.

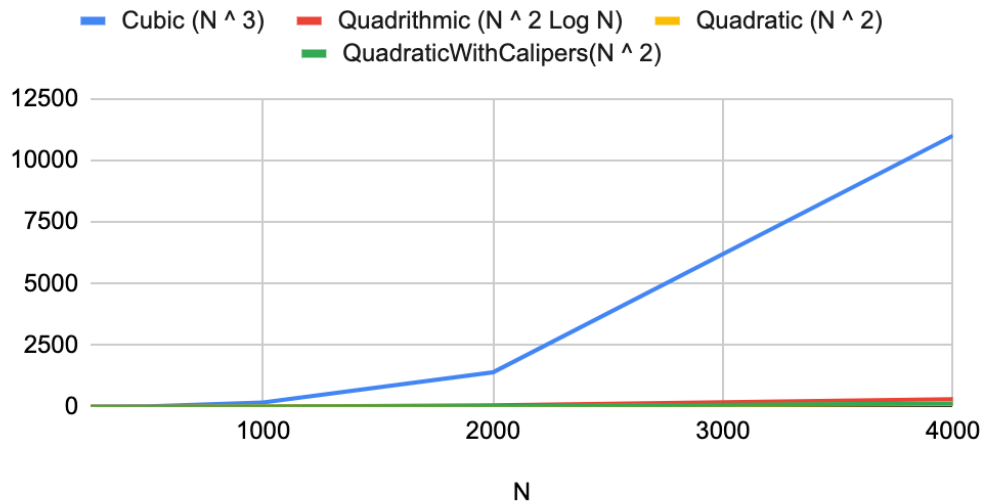
Additionally, the two-pointer method allows us to quickly discard any pairs of numbers that are not a valid solution, as soon as we find that their sum is either greater or less than the target value, this is why it's more efficient than the cubic method.

Evidence to support that conclusion:

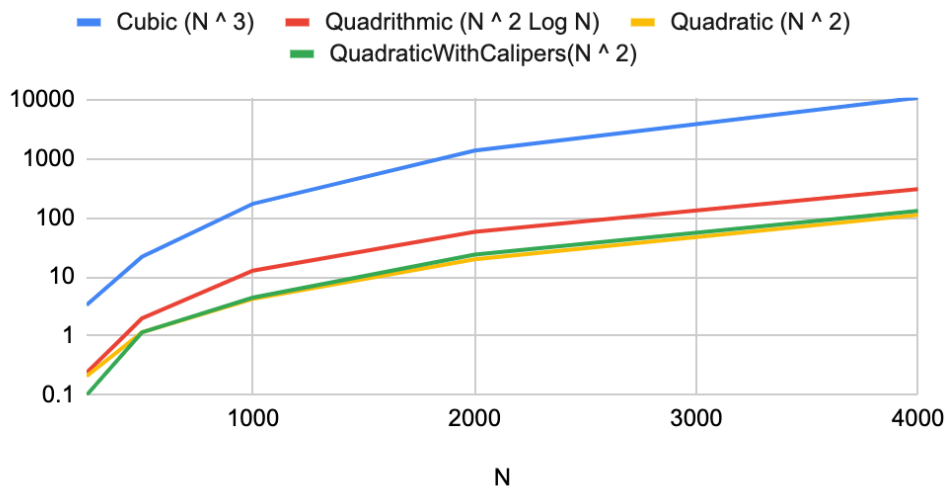
N	Cubic (N^3)	Quadrithmic ($N^2 \log N$)	Quadratic (N^2)	QuadraticWithCalipers(N^2)
250	3.36	0.24	0.21	0.1
500	22.08	2	1.16	1.16
1000	174.4	12.9	4.3	4.5
2000	1397.6	58.4	20	24.1
4000	11005.6	311	113.2	132.2

Graphical Representation:

Line Chart Comparison



Logarithmic Chart Comparison

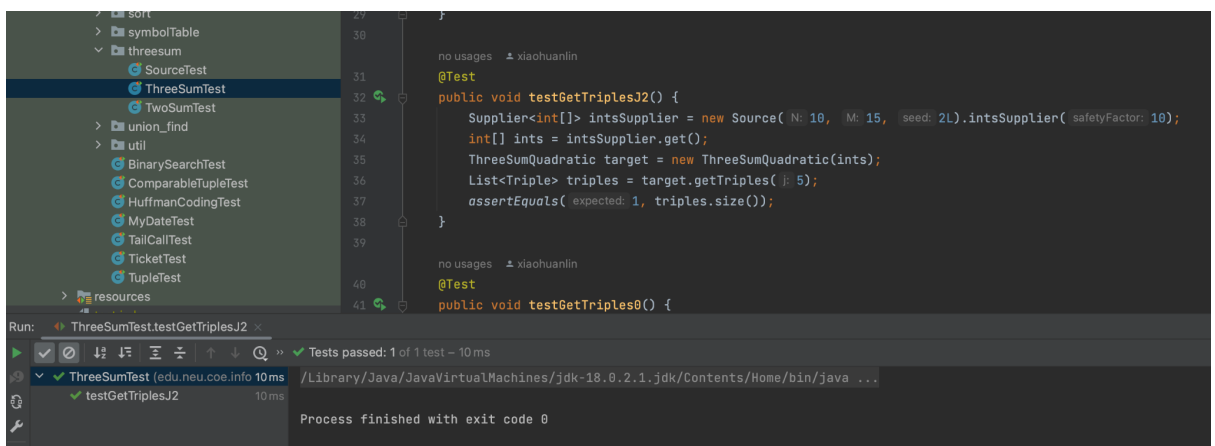
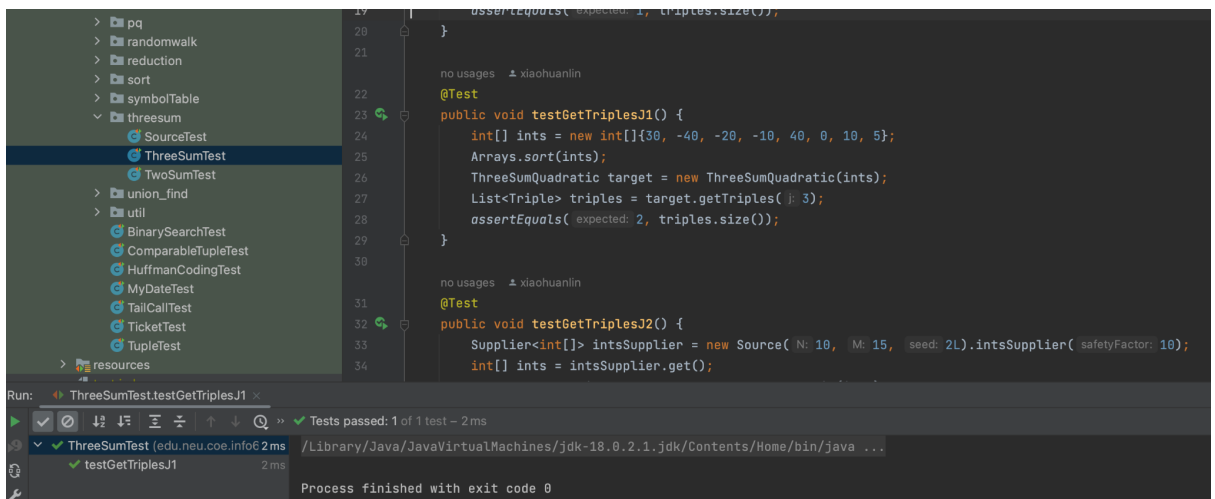
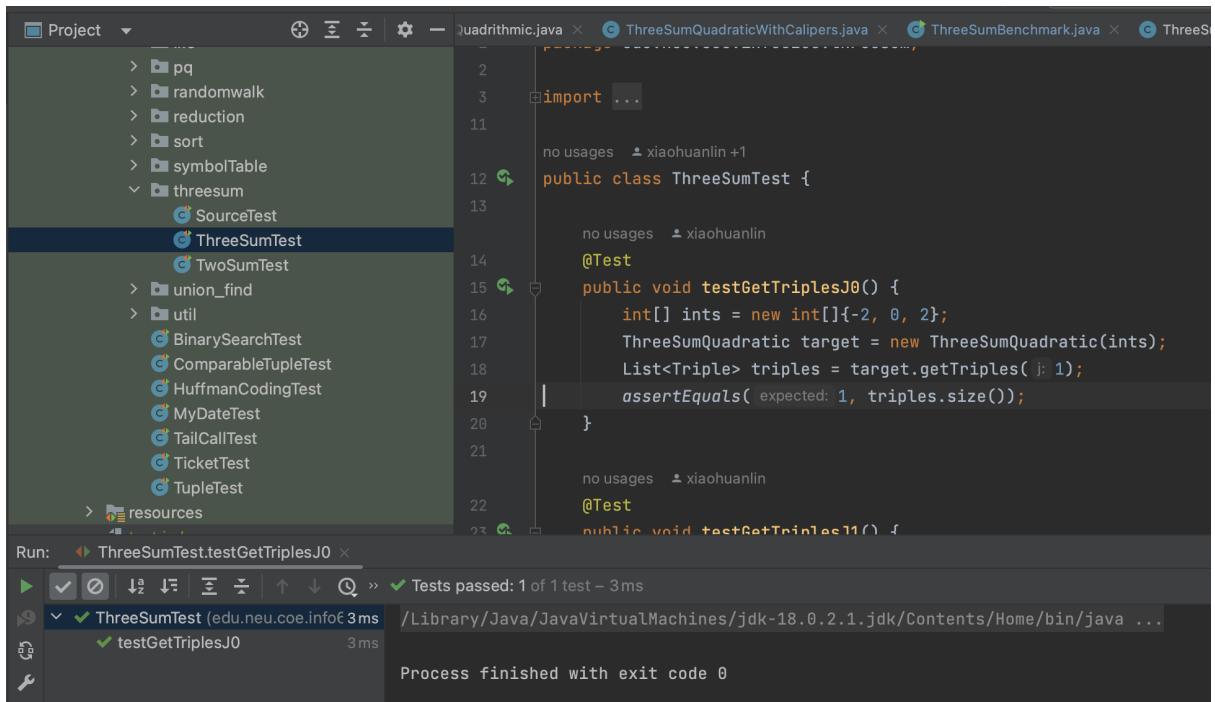


Unit Test Screenshots:

```
Run: ThreeSumTest
Tests passed: 11 of 11 tests - 790 ms

ThreeSumTest (edu.neu.coe.inf.790 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
  testGetTriples0 7 ms ints: [-40, -20, -10, 0, 5, 10, 30, 40]
  testGetTriples1 3 ms 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}]
  testGetTriples2 0 ms triples: [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}]
  testGetTriplesC0 0 ms [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}]
  testGetTriplesC1 1 ms [-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
  testGetTriplesC2 1 ms [Triple{x=-29, y=5, z=24}]
  testGetTriplesC3 202 ms ints: [-40, -20, -10, 0, 5, 10, 30, 40]
  testGetTriplesC4 678 ms 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}]
  testGetTriplesJ0 0 ms [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}]
  testGetTriplesJ1 1 ms [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}]
  testGetTriplesJ2 0 ms [-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

```

```

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

```

```

@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 x

Tests passed: 1 of 1 test - 9 ms

ThreeSumTest (edu.neu.coe.info 9 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...

testGetTriples0 9 ms

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

```

> sort
> symbolTable
> threesum
  SourceTest
  ThreeSumTest
  TwoSumTest
> union_find
> util
  BinarySearchTest
  ComparableTupleTest
  HuffmanCodingTest
  MyDateTest
  TailCallTest
  TicketTest
  TupleTest
> resources

```

```

52
53
54
55
56
57
58
59
60
61
62
63
64

```

```

@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);
}

```

no usages xiaohuanlin

Run: ThreeSumTest.testGetTriples1 x

Tests passed: 1 of 1 test - 12 ms

ThreeSumTest (edu.neu.coe.info 12 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...

testGetTriples1 12 ms

[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

```

```

65
66
67
68
69
70
71
72
73
74
75
76
77

```

```

@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 x

Tests passed: 1 of 1 test - 14 ms

ThreeSumTest (edu.neu.coe.info 14 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...

testGetTriples2 14 ms

[-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

```

```

99
100 @Test
101 public void testGetTriplesC0() {
102     int[] ints = new int[]{30, -40, -20, -10, 40, 0, 10, 5};
103     Arrays.sort(ints);
104     System.out.println("ints: " + Arrays.toString(ints));
105     ThreeSum target = new ThreeSumQuadratic(ints);
106     Triple[] triples = target.getTriples();
107     System.out.println("triples: " + Arrays.toString(triples));
108     assertEquals( expected: 4, triples.length);
109     assertEquals( expected: 4, new ThreeSumCubic(ints).getTriples().length);
110 }
111
112 no usages  xiaohuanlin +1
113 @Test
114 public void testGetTriplesC1() {
115     Supplier<int[]> intsSupplier = new Source( N: 20, M: 20, seed: 1L).intsSupplier( safetyFactor: 10);

```

Run: ThreeSumTest.testGetTriplesC0 x

Tests passed: 1 of 1 test - 10 ms

ThreeSumTest (edu.neu.coe.info 10 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...

testGetTriplesC0 10 ms

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

```

```

111 no usages  xiaohuanlin +1
112 @Test
113 public void testGetTriplesC1() {
114     Supplier<int[]> intsSupplier = new Source( N: 20, M: 20, seed: 1L).intsSupplier( safetyFactor: 10);
115     int[] ints = intsSupplier.get();
116     ThreeSum target = new ThreeSumQuadraticWithCalipers(ints);
117     Triple[] triples = target.getTriples();
118     assertEquals( expected: 4, triples.length);
119     System.out.println(Arrays.toString(triples));
120     Triple[] triples2 = new ThreeSumCubic(ints).getTriples();
121     System.out.println(Arrays.toString(triples2));
122     assertEquals( expected: 4, triples2.length);
123 }
124
125 no usages  xiaohuanlin +1
126 @Test

```

Run: ThreeSumTest.testGetTriplesC0 x

Tests passed: 1 of 1 test - 10 ms

ThreeSumTest (edu.neu.coe.info 10 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...

testGetTriplesC0 10 ms

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

```

```

124
125 @Test
126 public void testGetTriplesC2() {
127     Supplier<int[]> intsSupplier = new Source( N: 10, M: 15, seed: 3L).intsSupplier( safetyFactor: 10);
128     int[] ints = intsSupplier.get();
129     ThreeSum target = new ThreeSumQuadraticWithCalipers(ints);
130     System.out.println(Arrays.toString(ints));
131     Triple[] triples = target.getTriples();
132     System.out.println(Arrays.toString(triples));
133     assertEquals( expected: 1, triples.length);
134     assertEquals( expected: 1, new ThreeSumCubic(ints).getTriples().length);
135 }
136
137 no usages  xiaohuanlin +1
138 @Test
139 public void testGetTriplesC3() {
140     Supplier<int[]> intsSupplier = new Source( N: 1000, M: 1000, seed: 1000L).intsSupplier( safetyFactor: 10);

```

Run: ThreeSumTest.testGetTriplesC2 x

Tests passed: 1 of 1 test - 15 ms

ThreeSumTest (edu.neu.coe.info 15 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...

testGetTriplesC2 15 ms

ints: [-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
triples: [Triple{x=-29, y=5, z=24}]

Process finished with exit code 0

