Program Structures and Algorithms
Spring 2024

NAME: Ting Guo
NUID: 002834835
GITHUB LINK:https://github.com/Nangongnuanshan/INFO6205

Task:Assignment 2

Explantion:

Quadratic:As stated in the code comments, this approach divides the solution-space into N sub-spaces where each sub-space corresponds to a fixed value for the middle index of the three values. Each sub-space is then solved by expanding the scope of the other two indices outwards from the starting point.
We need to solve two problems to ensure that this method is correct.
First, make sure that the sum of the three numbers found by this method is 0.
That's easy. We know j is the index of the middle value. i < j < k. i = 0, k = length – 1.We just need make sure a[i] + a[j] + a[k] == 0.

```
while (i < j && j < k) {
    int sum = a[i] + a[j] + a[k];
    if (sum == 0) {
        triples.add(new Triple(a[i], a[j], a[k]));
        i++;
        k--;
```

Second, make sure we would not lose any answer.
First, use for() to make sure we use each element in int[] a as the middle value

```
for (int i = 0; i < length; i++) triples.addAll(getTriples(i));
```

Second, when we choose a j, make sure we find each i and k and check a[i] + a[j] + a[k] == 0

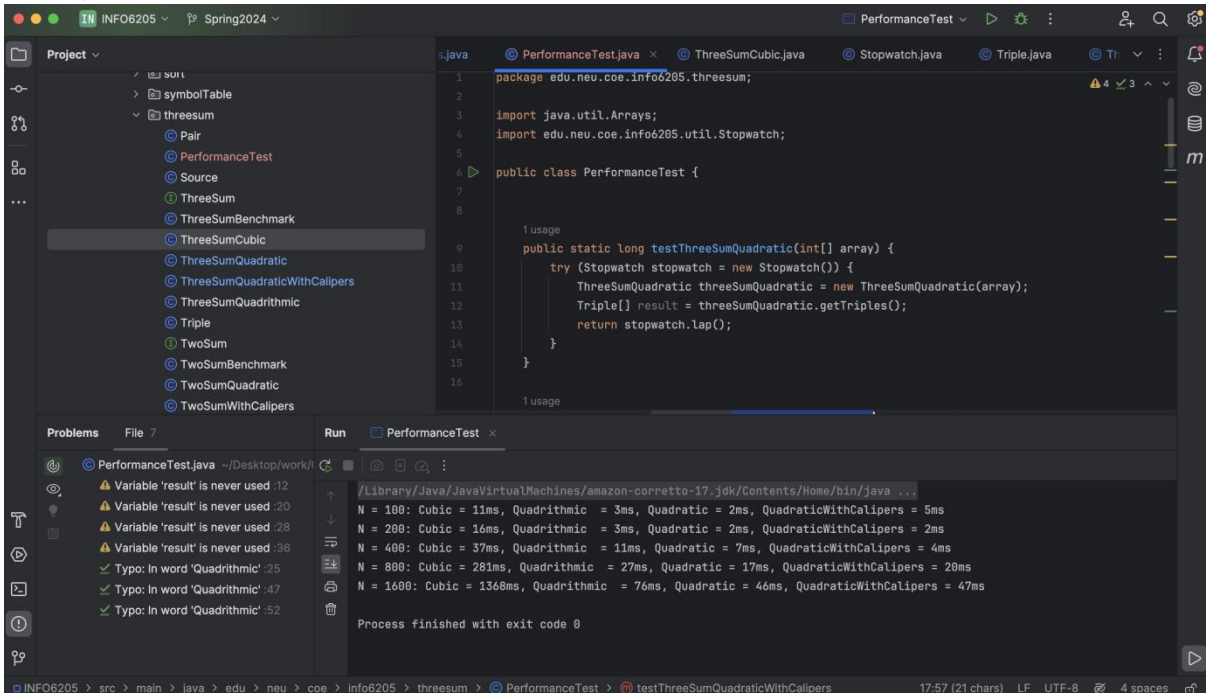
```
} else if (sum < 0) {
    i++;
    while (i < j && a[i] == a[i - 1]) i++;
} else {
    k--;
    while (j < k && a[k] == a[k + 1]) k--;
}
```

QuadraticWithCalipers: Same as Quadratic. The only different from Quadratic is that we use the first element instead of the middle one, which means we know the minimum in three. What we need change is that left = i, right = length – 1.(in this code , the first element is i) and while(left < right). We need to find two numbers from A to B, so that the sum of them is –i. We use left++ and right–– to make sure we will not skip any element.

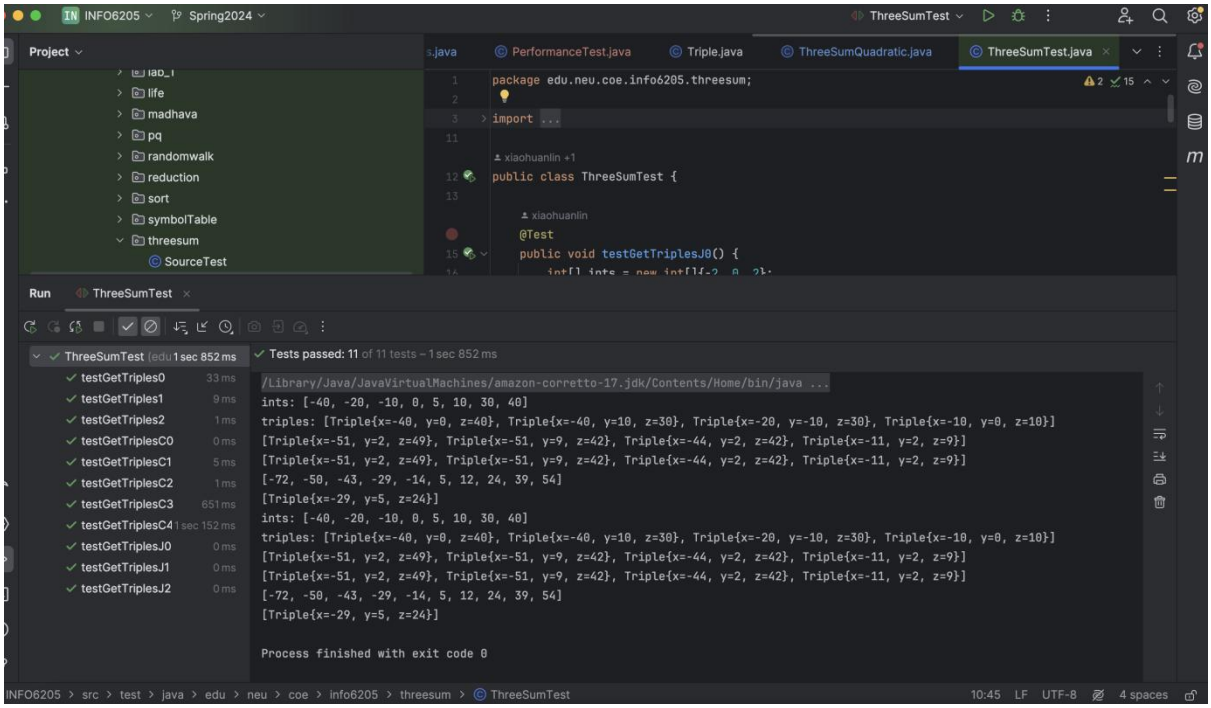
```
while (left < right) {
    Triple triple = new Triple(a[i], a[left], a[right]);
    int sum = function.apply(triple);
    if (sum == 0) {
        triples.add(triple);
        left++;
        right--;
```

The reason that left starts at i+1 is left > i, and because i starts at 0, we can make sure that We have checked all combinations (left < i+1). When left < i +1, we can see left as i and i as left, we will know we have check this combination.

Timing observations:



Unit Test Screenshots:


INFO6205
Spring2024
ThreeSumTest
Project
lab_1
life
madhava
pq
randomwalk
reduction
sort
symbolTable
threesum
SourceTest
s.java
PerformanceTest.java
Triple.java
ThreeSumQuadratic.java
ThreeSumTest.java
package edu.neu.coe.info6205.threesum;
import ...
xiaohuanlin +1
public class ThreeSumTest {
xiaohuanlin
@Test
public void testGetTriplesJ0() {
Run
ThreeSumTest
ThreeSumTest (edu 1 sec 852 ms
testGetTriples0 33 ms
testGetTriples1 9 ms
testGetTriples2 1 ms
testGetTriplesC0 0 ms
testGetTriplesC1 5 ms
testGetTriplesC2 1 ms
testGetTriplesC3 651 ms
testGetTriplesC4 1 sec 152 ms
testGetTriplesJ0 0 ms
testGetTriplesJ1 0 ms
testGetTriplesJ2 0 ms
Tests passed: 11 of 11 tests – 1 sec 852 ms
/Library/Java/JavaVirtualMachines/amazon-corretto-17.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}]
[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}]
[-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
[Triple{x=-29, y=5, z=24}]
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}]
[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}]
[-72, -50, -43, -29, -14, 5, 12, 24, 39, 54]
[Triple{x=-29, y=5, z=24}]
Process finished with exit code 0
INFO6205 > src > test > java > edu > neu > coe > info6205 > threesum > ThreeSumTest
10:45 LF UTF-8 4 spaces