Program Structures and Algorithms

Spring 2024

NAME: Ting Guo

NUID: 002834835

GITHUB LINK:https://github.com/Nangongnuanshan/INFO6205

**Task:Assignment 2**

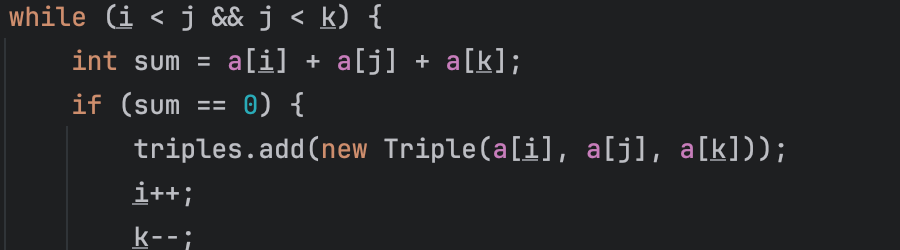
**Explantion:**

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.

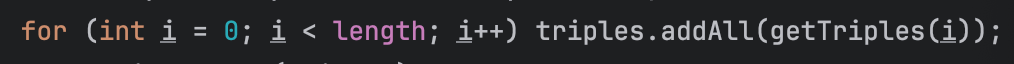
First, make sure that the sum of the three numbers found by this method is 0.

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.

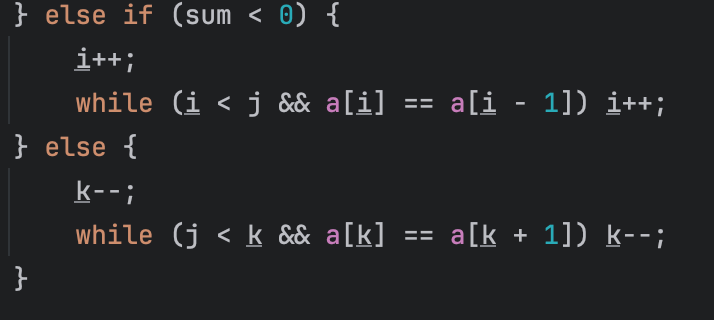


Second, make sure we would not lose any answer.

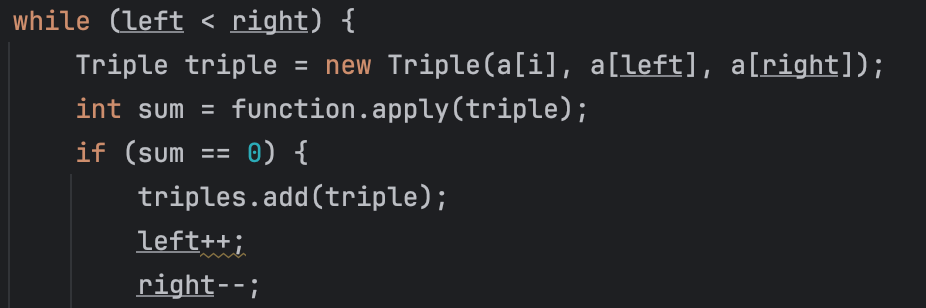
First, use for() to make sure we use each element in int[] a as the middle value



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

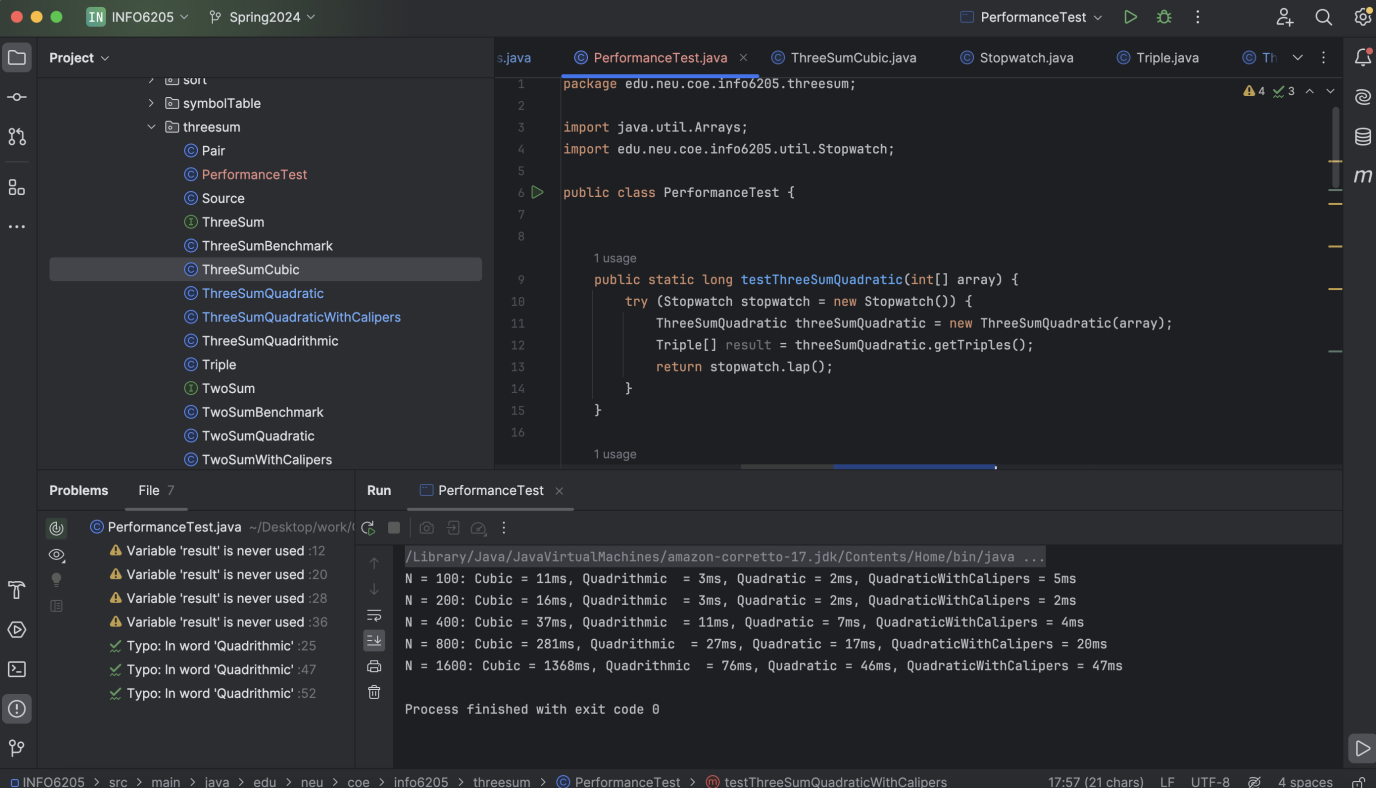


QuadraticWithCalipers: Same as Quadratic. The only different from Quadratic is that we use the first element instead of the middle one, which means we know the minimum in three. What we need change is that left = i, right = length - 1.(in this code , the first element is i) and while(left < right). We 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.



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:**

