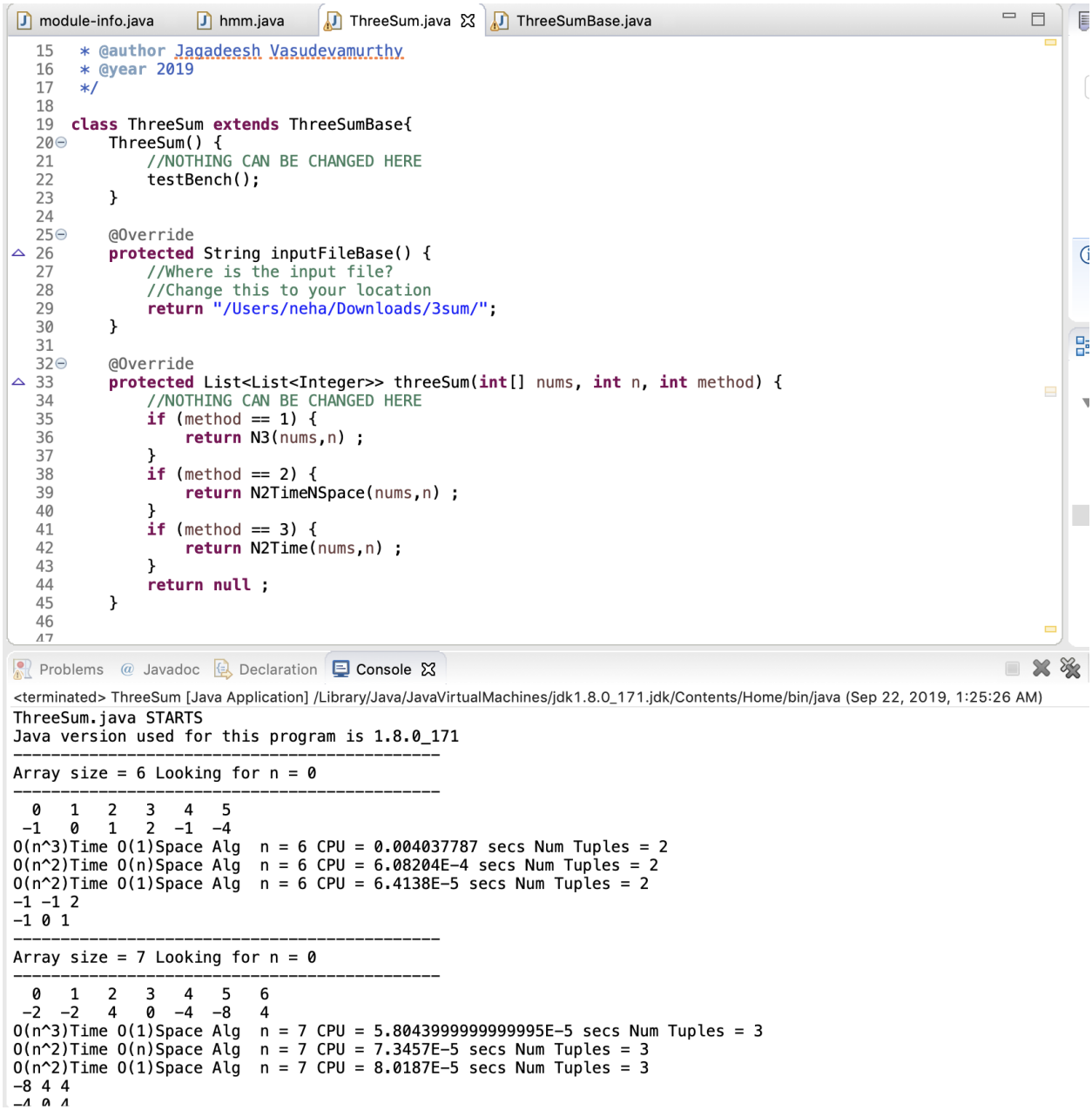
Output:

```
ThreeSum.java STARTS
Java version used for this program is 1.8.0_171
-------------------------------------------
Array size = 6 Looking for n = 0
-------------------------------------------
  0   1   2   3   4   5
 -1   0   1   2  -1  -4
O(n^3)Time O(1)Space Alg  n = 6 CPU = 0.004037787 secs Num Tuples = 2
O(n^2)Time O(n)Space Alg  n = 6 CPU = 6.08204E-4 secs Num Tuples = 2
O(n^2)Time O(1)Space Alg  n = 6 CPU = 6.4138E-5 secs Num Tuples = 2
-1 -1 2
-1 0 1
-------------------------------------------
Array size = 7 Looking for n = 0
-------------------------------------------
  0   1   2   3   4   5   6
 -2  -2   4   0  -4  -8   4
O(n^3)Time O(1)Space Alg  n = 7 CPU = 5.8043999999999995E-5 secs Num Tuples = 3
O(n^2)Time O(n)Space Alg  n = 7 CPU = 7.3457E-5 secs Num Tuples = 3
O(n^2)Time O(1)Space Alg  n = 7 CPU = 8.0187E-5 secs Num Tuples = 3
-8 4 4
-4 0 4
-2 -2 4
-------------------------------------------
Array size = 7 Looking for n = 0
-------------------------------------------
  0   1   2   3   4   5   6
 -1   0   1   0   1   0   1
O(n^3)Time O(1)Space Alg  n = 7 CPU = 1.6018E-4 secs Num Tuples = 2
O(n^2)Time O(n)Space Alg  n = 7 CPU = 1.32349E-4 secs Num Tuples = 2
O(n^2)Time O(1)Space Alg  n = 7 CPU = 7.4843E-5 secs Num Tuples = 2
-1 0 1
0 0 0
-------------------------------------------
Array size = 8 Looking for n = 0
-------------------------------------------
  0   1   2   3   4   5   6   7
 30 -30 -20 -10  40   0  10  15
O(n^3)Time O(1)Space Alg  n = 8 CPU = 5.5898999999999995E-5 secs Num Tuples = 4
O(n^2)Time O(n)Space Alg  n = 8 CPU = 6.8082E-5 secs Num Tuples = 4
O(n^2)Time O(1)Space Alg  n = 8 CPU = 5.13E-5 secs Num Tuples = 4
-30 -10 40
-30 0 30
-20 -10 30
-10 0 10
-------------------------------------------
Array size = 5 Looking for n = 0
-------------------------------------------
  0   1   2   3   4
 -2   0   1   1   2
O(n^3)Time O(1)Space Alg  n = 5 CPU = 2.5628000000000003E-5 secs Num Tuples = 2
O(n^2)Time O(n)Space Alg  n = 5 CPU = 3.3853E-5 secs Num Tuples = 2
```

```
O(n^2)Time O(1)Space Alg  n = 5 CPU = 2.6625E-5 secs Num Tuples = 2
-2 0 2
-2 1 1
-------------------------------------------
Array size = 6 Looking for n = 0
-------------------------------------------
  0   1   2   3   4   5
-25069 -9243 34312 -55692 -9243 64935
O(n^3)Time O(1)Space Alg  n = 6 CPU = 4.6563E-5 secs Num Tuples = 2
O(n^2)Time O(n)Space Alg  n = 6 CPU = 6.7038E-5 secs Num Tuples = 2
O(n^2)Time O(1)Space Alg  n = 6 CPU = 3.4368999999999996E-5 secs Num Tuples = 2
-55692 -9243 64935
-25069 -9243 34312
-------------------------------------------
Array size = 2000 Looking for n = 0
-------------------------------------------
O(n^3)Time O(1)Space Alg  n = 2000 CPU = 1.869928456 secs Num Tuples = 528
O(n^2)Time O(n)Space Alg  n = 2000 CPU = 0.085253377 secs Num Tuples = 528
O(n^2)Time O(1)Space Alg  n = 2000 CPU = 0.055526547999999995 secs Num Tuples = 528
-------------------------------------------
Array size = 4000 Looking for n = 0
-------------------------------------------
O(n^3)Time O(1)Space Alg  n = 4000 CPU = 14.652667745999999 secs Num Tuples = 2
O(n^2)Time O(n)Space Alg  n = 4000 CPU = 0.119701796 secs Num Tuples = 2
O(n^2)Time O(1)Space Alg  n = 4000 CPU = 0.14944898599999998 secs Num Tuples = 2
-1906468524 308903591 1597564933
-1825991740 426869298 1399122442
-------------------------------------------
Array size = 8000 Looking for n = 0
-------------------------------------------
O(n^3)Time O(1)Space Alg  n = 8000 CPU = 150.437339554 secs Num Tuples = 18
O(n^2)Time O(n)Space Alg  n = 8000 CPU = 6.596547882000001 secs Num Tuples = 18
O(n^2)Time O(1)Space Alg  n = 8000 CPU = 0.6200709419999999 secs Num Tuples = 18
-2060009284 770501022 1289508262
-1988258329 -1862536728 -444172239
-1965303896 510253452 1455050444
-1872625053 676494276 1196130777
-1747090115 -1741658624 -806218557
-1692722126 -1411160544 -1191084626
-1670186372 -12384429 1682570801
-1618442978 -28322173 1646765151
-1536320024 -400165924 1936485948
-1347507086 -311225880 1658732966
-1347507086 525082941 822424145
-1160601278 -401009979 1561611257
-1138819208 -855391919 1994211127
-551508475 215082405 336426070
247063352 1946436654 2101467290
667938470 1752128959 1874899867
802480953 1374354298 2118132045
1136081531 1217215378 1941670387
-------------------------------------------
Array size = 16000 Looking for n = 0
-------------------------------------------
O(n^2)Time O(n)Space Alg  n = 16000 CPU = 26.090184435000003 secs Num Tuples = 166
```

```
O(n^2)Time O(1)Space Alg  n = 16000 CPU = 2.666103491 secs Num Tuples = 166
-------------------------------------------
Array size = 128000 Looking for n = 0
-------------------------------------------
O(n^2)Time O(n)Space Alg  n = 128000 CPU = 980.974086541 secs Num Tuples = 81943
O(n^2)Time O(1)Space Alg  n = 128000 CPU = 243.050687748 secs Num Tuples = 81943
-------------------------------------------
Array size = 3000 Looking for n = 0
-------------------------------------------
O(n^3)Time O(1)Space Alg  n = 3000 CPU = 6.431265381999999 secs Num Tuples = 16258
O(n^2)Time O(n)Space Alg  n = 3000 CPU = 0.833761717 secs Num Tuples = 16258
O(n^2)Time O(1)Space Alg  n = 3000 CPU = 0.097597509 secs Num Tuples = 16258
All simple tests passed
All test cases passed. You will get full grade now
Include only ThreeSum.java and the output of this program for FULL grade
ThreeSum.java ENDS
```

ScreenShot:



module-info.java | hmm.java | ThreeSum.java | ThreeSumBase.java

```
 * @author Jagadeesh Vasudevamurthy
 * @year 2019
 */

class ThreeSum extends ThreeSumBase{
    ThreeSum() {
        //NOTHING CAN BE CHANGED HERE
        testBench();
    }

    @Override
    protected String inputFileBase() {
        //Where is the input file?
        //Change this to your location
        return "/Users/neha/Downloads/3sum/";
    }

    @Override
    protected List<List<Integer>> threeSum(int[] nums, int n, int method) {
        //NOTHING CAN BE CHANGED HERE
        if (method == 1) {
            return N3(nums,n) ;
        }
        if (method == 2) {
            return N2TimeNSpace(nums,n) ;
        }
        if (method == 3) {
            return N2Time(nums,n) ;
        }
        return null ;
    }

```

Problems | Javadoc | Declaration | Console

```
<terminated> ThreeSum [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_171.jdk/Contents/Home/bin/java (Sep 22, 2019, 1:25:26 AM)
ThreeSum.java STARTS
Java version used for this program is 1.8.0_171
-----------------------------------------------
Array size = 6 Looking for n = 0
-----------------------------------------------
  0   1   2   3   4   5
 -1   0   1   2  -1  -4
O(n^3)Time O(1)Space Alg  n = 6 CPU = 0.004037787 secs Num Tuples = 2
O(n^2)Time O(n)Space Alg  n = 6 CPU = 6.08204E-4 secs Num Tuples = 2
O(n^2)Time O(1)Space Alg  n = 6 CPU = 6.4138E-5 secs Num Tuples = 2
-1 -1 2
-1 0 1
-----------------------------------------------
Array size = 7 Looking for n = 0
-----------------------------------------------
  0   1   2   3   4   5   6
 -2  -2   4   0  -4  -8   4
O(n^3)Time O(1)Space Alg  n = 7 CPU = 5.80439999999999995E-5 secs Num Tuples = 3
O(n^2)Time O(n)Space Alg  n = 7 CPU = 7.3457E-5 secs Num Tuples = 3
O(n^2)Time O(1)Space Alg  n = 7 CPU = 8.0187E-5 secs Num Tuples = 3
-8 4 4
-4 0 4
```