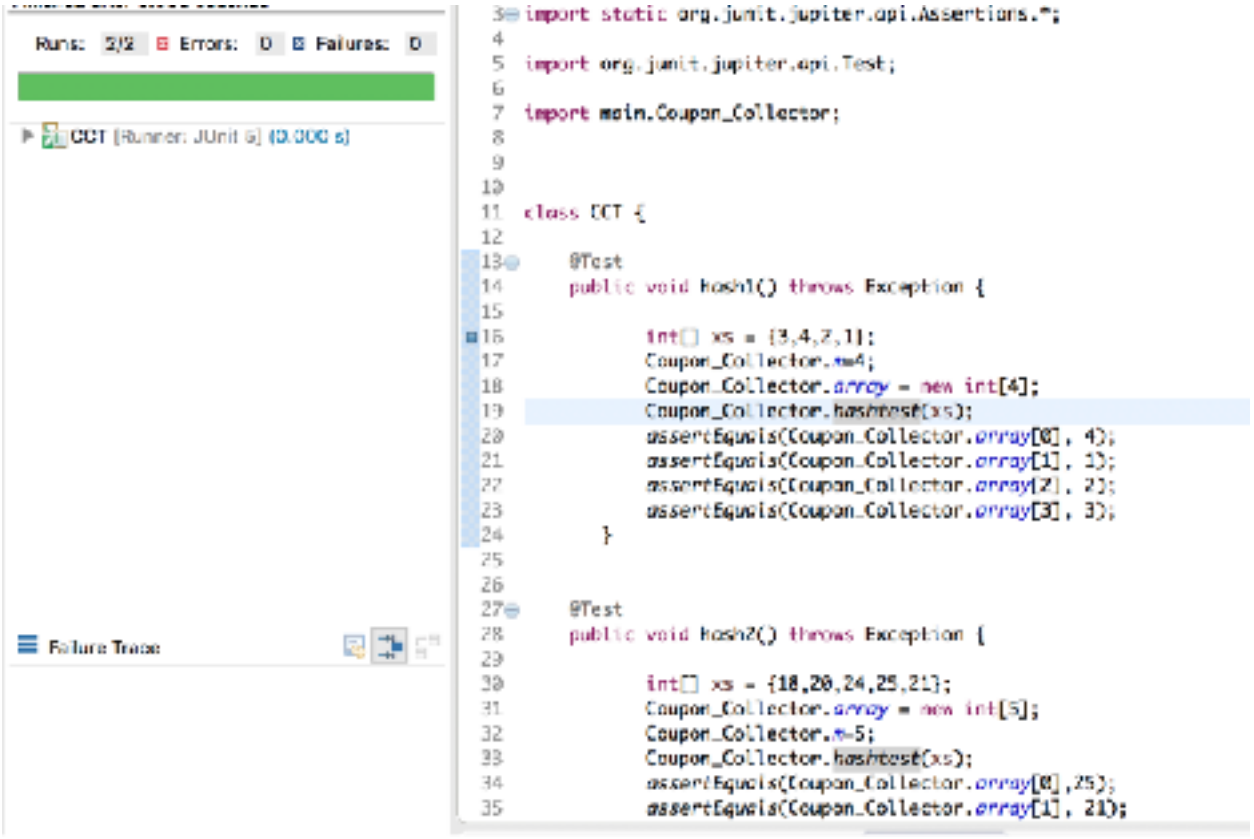
M value	C1(m) Progam (avg)Value	C1(m) Theoretical Value	B0(M) Program(avg) Value	B0(M) Theoretical Value
100	12.154	12.533	566.454	460.517
250	17.445	17.724	1259.855	1059.663
365	23.449	23.944	2525.89	2153.46
500	28.445	28.024	3609	3107.304
1000	39.01	39.63	7983.091	6907.755

Based on the above observations we notice that the theoretical value and programming values are mostly similar.

From the above results, it can be validated that the theoretical value for C1(m) ~ sqrt(pi * m/2)

And also for all the bins to be filled from the programming values it can be verified that B0(m) ~ m ln m

Test Cases:


Runs: 2/2 Errors: 0 Failures: 0
CCT [Runner: JUnit 5] (0.000 s)
Failure Trace
3 import static org.junit.jupiter.api.Assertions.*;
4
5 import org.junit.jupiter.api.Test;
6
7 import main.Coupon_Collector;
8
9
10
11 class CCT {
12
13 @Test
14 public void Hash1() throws Exception {
15
16 int[] xs = {3,4,2,1};
17 Coupon_Collector.m4;
18 Coupon_Collector.array = new int[4];
19 Coupon_Collector.hashtest(xs);
20 assertEquals(Coupon_Collector.array[0], 4);
21 assertEquals(Coupon_Collector.array[1], 1);
22 assertEquals(Coupon_Collector.array[2], 2);
23 assertEquals(Coupon_Collector.array[3], 3);
24 }
25
26
27 @Test
28 public void Hash2() throws Exception {
29
30 int[] xs = {18,20,24,25,21};
31 Coupon_Collector.array = new int[5];
32 Coupon_Collector.m=5;
33 Coupon_Collector.hashtest(xs);
34 assertEquals(Coupon_Collector.array[0],25);
35 assertEquals(Coupon_Collector.array[1], 21);