StrassenMatrixMultiplication.java

```
1   package DivideAndConquerTesting;
2
3   public class StrassenMatrixMultiplication {
4           public int[][] multiply(int[][] A, int[][] B) {
5           int n = A.length;
6
7           int[][] R = new int[n][n];
8
9  1        if (n == 1) {
10 1            R[0][0] = A[0][0] * B[0][0];
11          } else {
12 2            int[][] A11 = new int[n / 2][n / 2];
13 2            int[][] A12 = new int[n / 2][n / 2];
14 2            int[][] A21 = new int[n / 2][n / 2];
15 2            int[][] A22 = new int[n / 2][n / 2];
16 2            int[][] B11 = new int[n / 2][n / 2];
17 2            int[][] B12 = new int[n / 2][n / 2];
18 2            int[][] B21 = new int[n / 2][n / 2];
19 2            int[][] B22 = new int[n / 2][n / 2];
20
21 1            split(A, A11, 0, 0);
22 2            split(A, A12, 0, n / 2);
23 2            split(A, A21, n / 2, 0);
24 3            split(A, A22, n / 2, n / 2);
25
26 1            split(B, B11, 0, 0);
27 2            split(B, B12, 0, n / 2);
28 2            split(B, B21, n / 2, 0);
29 3            split(B, B22, n / 2, n / 2);
30
31              int[][] M1 = multiply(add(A11, A22), add(B11, B22));
32              int[][] M2 = multiply(add(A21, A22), B11);
33              int[][] M3 = multiply(A11, sub(B12, B22));
34              int[][] M4 = multiply(A22, sub(B21, B11));
35              int[][] M5 = multiply(add(A11, A12), B22);
36              int[][] M6 = multiply(sub(A21, A11), add(B11, B12));
37              int[][] M7 = multiply(sub(A12, A22), add(B21, B22));
38              int[][] C11 = add(sub(add(M1, M4), M5), M7);
39              int[][] C12 = add(M3, M5);
40              int[][] C21 = add(M2, M4);
41              int[][] C22 = add(sub(add(M1, M3), M2), M6);
```

```
42
43 1            join(C11, R, 0, 0);
44 2            join(C12, R, 0, n / 2);
45 2            join(C21, R, n / 2, 0);
46 3            join(C22, R, n / 2, n / 2);
47          }
48 1        return R;
49      }
50
51      public int[][] sub(int[][] A, int[][] B) {
52          int n = A.length;
53          int[][] C = new int[n][n];
54 3        for (int i = 0; i < n; i++) {
55 3            for (int j = 0; j < n; j++) {
56 1                C[i][j] = A[i][j] - B[i][j];
57              }
58          }
59 1        return C;
60      }
61
62      public int[][] add(int[][] A, int[][] B) {
63          int n = A.length;
64          int[][] C = new int[n][n];
65
66 3        for (int i = 0; i < n; i++) {
67 3            for (int j = 0; j < n; j++) {
68 1                C[i][j] = A[i][j] + B[i][j];
69              }
70          }
71 1        return C;
72      }
73
74      public void split(int[][] P, int[][] C, int iB, int jB) {
75 4        for (int i1 = 0, i2 = iB; i1 < C.length; i1++, i2++) {
76 4            for (int j1 = 0, j2 = jB; j1 < C.length; j1++, j2++) {
77                  C[i1][j1] = P[i2][j2];
78              }
79          }
80      }
81
82      public void join(int[][] C, int[][] P, int iB, int jB) {
83 4        for (int i1 = 0, i2 = iB; i1 < C.length; i1++, i2++) {
84 4            for (int j1 = 0, j2 = jB; j1 < C.length; j1++, j2++) {
85                  P[i2][j2] = C[i1][j1];
86              }
```

```
87          }
88      }
89  }
```

Mutations

9 1. negated conditional → KILLED

10 1. Replaced integer multiplication with division → KILLED

12 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → SURVIVED

13 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → SURVIVED

14 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → SURVIVED

15 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → SURVIVED

16 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → SURVIVED

17 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → SURVIVED

18 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → SURVIVED

19 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → SURVIVED

21 1. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::split → KILLED

22 1. Replaced integer division with multiplication → KILLED
2. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::split → KILLED

23 1. Replaced integer division with multiplication → KILLED
2. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::split → KILLED

24 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → KILLED
3. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::split → KILLED

26 1. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::split → KILLED

27 1. Replaced integer division with multiplication → KILLED
2. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::split → KILLED

28 1. Replaced integer division with multiplication → KILLED
2. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::split → KILLED

29 1. Replaced integer division with multiplication → KILLED
2. Replaced integer division with multiplication → KILLED
3. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::split → KILLED

43 1. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::join → KILLED

44	1. Replaced integer division with multiplication → KILLED 2. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::join → KILLED
45	1. Replaced integer division with multiplication → KILLED 2. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::join → KILLED
46	1. Replaced integer division with multiplication → KILLED 2. Replaced integer division with multiplication → KILLED 3. removed call to DivideAndConquerTesting/StrassenMatrixMultiplication::join → KILLED
48	1. replaced return value with null for DivideAndConquerTesting/StrassenMatrixMultiplication::multiply → KILLED
54	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
55	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
56	1. Replaced integer subtraction with addition → KILLED
59	1. replaced return value with null for DivideAndConquerTesting/StrassenMatrixMultiplication::sub → KILLED
66	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
67	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
68	1. Replaced integer addition with subtraction → KILLED
71	1. replaced return value with null for DivideAndConquerTesting/StrassenMatrixMultiplication::add → KILLED
75	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. Changed increment from 1 to -1 → KILLED 4. negated conditional → KILLED
76	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. Changed increment from 1 to -1 → KILLED 4. negated conditional → KILLED
83	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. Changed increment from 1 to -1 → KILLED 4. negated conditional → KILLED
84	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. Changed increment from 1 to -1 → KILLED 4. negated conditional → KILLED

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN

- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- DivideAndConquerTesting.AllDivideConquerTesting.[engine:junit-jupiter]/[class:DivideAndConquerTesting.AllDivideConquerTesting]/[method:testStrassenMatrixMultiplication()] (13 ms)
- DivideAndConquerTesting.AllDivideConquerTesting.[engine:junit-jupiter]/[class:DivideAndConquerTesting.AllDivideConquerTesting]/[method:BinarySearch2dArrayTestMiddle()] (28 ms)

Report generated by PIT 1.6.8