将边分为>1,=1,=0,<0四种情况

@Test

public void testTriang\_1(){

assertEquals(4,tri.Triang(2,2,2));

assertEquals(4,tri.Triang(1,1,1));

assertEquals(4,tri.Triang(0,0,0));

assertEquals(4,tri.Triang(-1,-1,-1));

}

@Test

public void testTriang\_2(){

assertEquals(4,tri.Triang(2,2,2));

assertEquals(4,tri.Triang(2,1,1));

assertEquals(4,tri.Triang(2,0,0));

assertEquals(4,tri.Triang(2,-1,-1));

assertEquals(4,tri.Triang(1,2,1));

assertEquals(4,tri.Triang(1,1,0));

assertEquals(4,tri.Triang(1,0,-1));

assertEquals(4,tri.Triang(1,-1,0));

assertEquals(4,tri.Triang(0,2,0));

assertEquals(4,tri.Triang(0,1,-1));

assertEquals(4,tri.Triang(0,0,2));

assertEquals(4,tri.Triang(0,-1,1));

assertEquals(4,tri.Triang(-1,2,-1));

assertEquals(4,tri.Triang(-1,1,2));

assertEquals(4,tri.Triang(-1,0,1));

assertEquals(4,tri.Triang(-1,-1,0));

}

@Test

public void testTriang\_3(){

assertEquals(1,tri.Triang(2,3,4));

assertEquals(2,tri.Triang(3,3,4));

assertEquals(3,tri.Triang(3,3,3));

assertEquals(4,tri.Triang(1,2,7));

assertEquals(4,tri.Triang(-1,2,7));

}

@Test

public void testTriang\_4(){

assertEquals(1,tri.Triang(2,3,4));

assertEquals(2,tri.Triang(3,3,4));

assertEquals(2,tri.Triang(3,4,3));

assertEquals(2,tri.Triang(4,3,3));

assertEquals(3,tri.Triang(3,3,3));

assertEquals(4,tri.Triang(1,2,7));

assertEquals(4,tri.Triang(1,7,2));

assertEquals(4,tri.Triang(7,2,1));

assertEquals(4,tri.Triang(2,2,7));

assertEquals(4,tri.Triang(2,7,2));

assertEquals(4,tri.Triang(7,2,2));

assertEquals(4,tri.Triang(-1,2,7));

assertEquals(4,tri.Triang(1,-2,7));

assertEquals(4,tri.Triang(1,2,-7));

}

@Test

public void testTriang\_5(){

assertEquals(1,tri.Triang(2,3,4));

assertEquals(2,tri.Triang(3,3,4));

assertEquals(3,tri.Triang(3,3,3));

assertEquals(4,tri.Triang(1,2,7));

}

@Test

public void testTriang\_6(){

//triOut = 4

assertEquals(4,tri.Triang(1,2,7));

assertEquals(4,tri.Triang(1,7,2));

assertEquals(4,tri.Triang(7,2,1));

assertEquals(4,tri.Triang(2,2,7));

assertEquals(4,tri.Triang(2,7,2));

assertEquals(4,tri.Triang(7,2,2));

assertEquals(4,tri.Triang(-1,2,7));

assertEquals(4,tri.Triang(1,-2,7));

assertEquals(4,tri.Triang(1,2,-7));

}

@Test

public void testTriangMutantOne(){

assertEquals(4,triMt1.Triang(2,2,7));

}

@Test

public void testTriangMutantTwo(){

assertEquals(4,triMt2.Triang(2,2,4));

**}**