

Test Case
BinarySearchTreeAVL.java

Single Right Rotation

Case 1 : single right rotation with replacing root `int arr1[] = {7, 5, 9, 3, 6, 1};`

Before rotation	After rotation	Pass/Fail
<pre> graph TD 7((7)) --- 5((5)) 7 --- 9((9)) 5 --- 3((3)) 5 --- 6((6)) 3 --- 1((1)) </pre>	<pre> graph TD 5((5)) --- 3((3)) 5 --- 7((7)) 3 --- 1((1)) 7 --- 6((6)) 7 --- 9((9)) </pre>	OK

Case 2 : single right rotation with replacing root `int arr2[] = {9, 10, 7, 8, 6, 5};`

Before rotation	After rotation	Pass/Fail
<pre> graph TD 9((9)) --- 7((7)) 9 --- 10((10)) 7 --- 6((6)) 7 --- 8((8)) 6 --- 5((5)) </pre>	<pre> graph TD 7((7)) --- 6((6)) 7 --- 9((9)) 6 --- 5((5)) 9 --- 8((8)) 9 --- 10((10)) </pre>	OK

Case 3 : single right rotation without replacing root `int arr3[] = {18, 20, 12, 22, 19, 14, 8, 9, 4, 2};`

Before rotation	After rotation	Pass/Fail
<pre> graph TD 18((18)) --- 12((12)) 18 --- 20((20)) 12 --- 8((8)) 12 --- 14((14)) 8 --- 4((4)) 8 --- 9((9)) 4 --- 2((2)) 20 --- 19((19)) 20 --- 22((22)) </pre>	<pre> graph TD 18((18)) --- 8((8)) 18 --- 20((20)) 8 --- 4((4)) 8 --- 12((12)) 4 --- 2((2)) 12 --- 9((9)) 12 --- 14((14)) 20 --- 19((19)) 20 --- 22((22)) </pre>	OK

Test Case
BinarySearchTreeAVL.java

Case 4 : single right rotation without replacing root `int arr4[] = {25, 27, 21, 30, 26, 22, 18, 20, 16, 12, 8};`

Before rotation	After rotation	Pass/Fail
		Pass/Fail NG To be revised

Single Left Rotation

Case 5 : single left rotation with replacing root `int arr5[] = {7, 9, 6, 12, 8, 14};`

Before rotation	After rotation	Pass/Fail
		Pass/Fail OK

Test Case
BinarySearchTreeAVL.java

Case 6 : single left rotation with replacing root `int arr6[] = {11, 14, 8, 19, 12, 20};`

Before rotation	After rotation	Pass/Fail
		OK

Case 7 : single left rotation without replacing root `int arr7[] = {26, 28, 23, 32, 27, 24, 19, 36, 30, 39};`

Before rotation	After rotation	Pass/Fail
		OK

Double Left Right Rotation **To be implemented**

Case 8 : double left right rotation `int arr8[] = {9, 11, 4, 6, 3, 8, 5};`

Before rotation	Left rotation (expected)	Right rotation (expected)	Pass/Fail
			NG

Test Case
BinarySearchTreeAVL.java

Double Right Left Rotation To be implemented

Case 9 : double right left rotation int arr9[] = {12, 17, 9, 20, 15, 16, 13};

Before rotation	Right rotation (expected)	Left rotation (expected)	Pass/Fail
<pre> graph TD 12((12)) --- 9((9)) 12 --- 17((17)) 17 --- 15((15)) 17 --- 20((20)) 15 --- 13((13)) 15 --- 16((16)) </pre>	<pre> graph TD 12((12)) --- 9((9)) 12 --- 15((15)) 15 --- 13((13)) 15 --- 17((17)) 17 --- 16((16)) 17 --- 20((20)) </pre>	<pre> graph TD 15((15)) --- 12((12)) 15 --- 17((17)) 12 --- 9((9)) 12 --- 13((13)) 17 --- 16((16)) 17 --- 20((20)) </pre>	NG