

Tracing: Searching through a BST

@Test
public void test_binary_search_trees_search() {
 BSTNode<String> n28 = new BSTNode<>(28, "alan");
 BSTNode<String> n21 = new BSTNode<>(21, "mark");
 BSTNode<String> n35 = new BSTNode<>(35, "tom");
 BSTNode<String> extN1 = new BSTNode<>();
 BSTNode<String> extN2 = new BSTNode<>();
 BSTNode<String> extN3 = new BSTNode<>();
 BSTNode<String> extN4 = new BSTNode<>();
 BSTNode<String> extN4 = new BSTNode<>();
 n28.setLeft(n21); n21.setParent(n28);
 n21.setLeft(extN1); extN1.setParent(n21);
 n21.setRight(extN2); extN2.setParent(n21);
 n35.setLeft(extN3); extN3.setParent(n35);
 n35.setRight(extN4); extN4.setParent(n35);

BSTUtilities<String> u = new BSTUtilities<>(); /* search existing keys */ assertTrue(n28 == u.search(n28, 28)); assertTrue(n21 == u.search(n28, 21)); assertTrue(n35 == u.search(n28, 35)); /* search non-existing keys */ assertTrue(extN1 == u.search(n28, 17)); /* *17* < 21 */ assertTrue(extN2 == u.search(n28, 23)); /* 21 < *23* < 28 */ assertTrue(extN3 == u.search(n28, 33)); /* 28 < *33* < 35 */ assertTrue(extN4 == u.search(n28, 38)); /* 35 < *38* */</pre>



Running Time: Search on a BST



Binary Search: Non-Linear vs. Linear Structures





Worst-Case RT: BST with Linear Height

Example 1: Inserted Entries with Decreasing Keys

<100, 75, 68, 60, 50, 1>

Example 2: Inserted Entries with Increasing Keys

<1, 50, 60, 68, 75, 100>

Example 3: Inserted Entries with In-Between Keys <1, 100, 50, 75, 60, 68>



Balanced BST: Definition



<u>Q</u>. Is the above tree a balanced BST?
<u>Q</u>. Still a balanced BST after inserting 55?
<u>Q</u>. Still a balanced BST after inserting 63?

What is a Priority Queue (PQ)

