Matt Ronan 4/3/20

Algorithms Assignment 7

Question 1 Section 1:

English: To find the kth smallest element, we must use an in order algorithm to find the kth smallest element, incrementing each time we check a node, then returning the value when our counter is equal to k.

Pseudocode:

public static Node kthSmallestElement(BST bst, int k)

int i = 0;

Node current = bst.root;

while(current != null)

if(current.left == null)

i++;

if(i == k)

return current;

else

current = current.right;

else

Node succ = current.left;

while (succ.right != null && succ.right != current)

succ = succ.left;

if (succ.right == null)

succ.right = current;

current = current.left;

else

succ.right = null;

i++;

if (i == k)

return current;

current = current.right;

return current;

Running Time: In a worst case scenario, this algorithm checks each node 1 time, resulting in a running time of O(n).

Section 1 Question 2:

English: We must recursively loop through the bst in an in order fashion, only printing the elements within the specified ranges.

Pseudocode:

public static void k1k2Range(Node r, int k1, int k2)

if (r != null)

k1k2Range(r.left, k1, k2);

if(r.data >= k1 && r.data <= k2) {

System.out.print(r.data+" ");

k1k2Range(r.right, k1, k2);

Running Time: In a worst case scenario, the user will specify a range of the entire tree, making the algorithm run in O(n).