# Michael Chillemi

# 06/16/2023

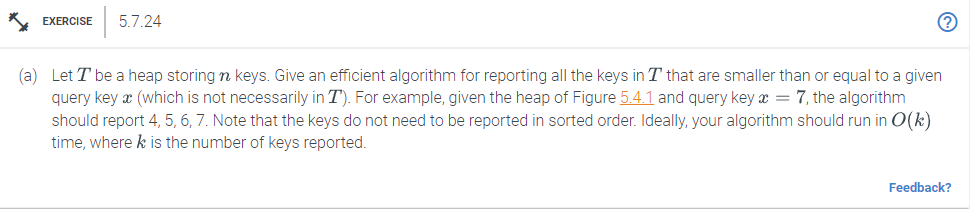
# CS 590 - Algorithms

# M4.B2: Module 4 Priority Queues and Heaps Creativity Exercises

# 

# 

Problem 5.7.24



Answer:

Algorithm listPreviousNodes(node,max\_key):

Input: node,max\_key

if (node == NULL)

return

if(node.data <= max\_key)

print(node.data)

listPreviousNodes(node.left,max\_key)

listPreviousNodes(node.right,max\_key)

else if(node.data > max\_key)

return

The way this algorithm works is first use the root node as input it will check if the node is null if it is then the function will just return. Next it will check if the current node's data is less than or equal to max\_num. If it is then it will print the current nodes data and then recursively call the function with the left and right nodes of the root, Then as a final check you must look out for a node’s data being greater than max\_num. If so then all use that as an indication to return the function. The algorithm provided above will have an O(k) time complexity. It is because k is the number of elements in the heap.