Question 3

**Input:**    two leftist heaps h1 and h2  
**Output:** leftist heaps resulting from the merge of h1 and h2

**Algorithm:**

**if**h1.key > h2.key **then**swap(h1,h2)

**if**h1.right=nil   
    **then** h1.right <-- h2  
    **else** h1.right <-- merge( h1.right, h2 )

**if** h1.left.rank < h1.right.rank  
    **then** swap( h1.left, h1.right )h1.rank <-- h1.right.rank + 1

**return** h1

public Node merge(Node x, Node y)

if (x == null)

return y;

if (y == null)

if (x.element.compareTo(y.element) > 0)

// x.element > y.element

Node temp = x;

x = y;

y = temp;

x.rightChild = merge(x.rightChild, y);

if (x.leftChild == null)

x.leftChild = x.rightChild;

x.rightChild = null;

else

if (x.leftChild.s < x.rightChild.s)

Node temp = x.leftChild;

x.leftChild = x.rightChild;

x.rightChild = temp;

x.s = x.rightChild.s + 1;

return x;

1. **The best case for this will O(log n)**
2. **For this will be insertion of nodes.**

**(c) The worst case time complexity is O(nlog *n*)**