**HW03-Joshua Godinez**

1. When a binary tree has a height of 0, it is only the root with no children and no internal nodes but is the only leaf. Let I= # internal node and L= # of leaf nodes. In this case, L=1 and I =0. Therefore L-1=I.
2. When a binary tree has each internal node containing two children it is considered a perfect binary tree. Let n = number of nodes and h= the height. Each level of the tree has twice as many nodes as the level before, therefore n=2k+1-1

n=2k+1-1 let k=0, n=20+1-1=1.

n=2k+1-1

n+1=2k+1

k+1=log2(n+1)

k=log2(n+1)-1 k is O(logn)

1. Let k be the height of the binary tree and n be the number of leaves.

A tree with height k would have 2k leaves. Therefore, n2k

n2k

n!2k

log2(n!)<=k

k=Ω(nlogn)

4. ) **QUICK SORT**

Y

NN

N

N

N

N

<a4,a3,a1,a2>

5.) **MERGE SORT**

<a4,a3,a1,a2>

NN

Y

NN

NN