3) Show the max number of nodes in a binary tree of height h is $2^{h+1}-1$.

Proof

Let $P(h) := 2^{h+1}-1$.

Base Case: $h=0$

When $h=0$, the tree has one node, the root meaning the max number of nodes is one. Thus,

$$\Rightarrow P(0) = 2^{0+1}-1 = 1$$
$$= 2^{1}-1 = 1$$
$$= 2-1 = 1$$
$$= 1 = 1.$$

Thus, the base case holds.

Inductive Step:

Let $P(K) := 2^{K+1} - 1$ the max number of nodes for a tree of hight K. Assume $P(K)$ is true. Then we will show a tree with arbitrary hight of $K+1$ has at most $P(K+1) = 2^{K+1+1} - 1 = 2^{k+2} - 1$ nodes (1).

By definition, the hight of a tree doesn't include the root and the max number of nods means both subtrees are full with hights of K.

Hence we have two subtrees of height K. Thus,

$\Rightarrow 2 \cdot (2^{K+1} - 1) + 1$ <+1 for the root>

$\Rightarrow 2^{K+1+1} - 2 + 1$

$= 2^{K+2} - 1$. (2)

Observe (1) = (2), thus our inductive hypothesis holds.

$\therefore$ The maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$.