CS 271 - Project 5

Nicholas Reichert, Oscar Martinez

October 23, 2019

1

Prove a complete binary tree with height h contains $2^{h+1} - 1$ total nodes.

Proof. We will prove this statement using induction.

Inductive Hypothesis: A complete binary tree with height h contains $2^{h+1} - 1$ total nodes.

Base Case: The simplest binary tree would be the binary tree with no nodes. However, that binary tree has no root, so its height is not defined. (The height of a tree is defined as the height of its root.)

The next simplest binary tree would be the binary tree with just a root. The tree would have height 0, since the root has no children.

$$2^{h+1} - 1 = 2^{0+1} - 1 = 2^1 - 1 = 2 - 1 = 1 \tag{1}$$

Our hypothesis for $h = 0$ is correct, because in total, this tree has 1 node.

Inductive Step:

Assume for the sake of induction that a complete binary tree with height h contains $2^{h+1} - 1$ total nodes.

Now, imagine a complete binary tree with height $h + 1$. It would have two children, each of which themselves are the roots of complete binary trees with height h. They each have $2^{h+1} - 1$ total nodes. The total number of nodes in this tree is

$$(2^{h+1} - 1) + (2^{h+1} - 1) + 1 \tag{2}$$

because the number of nodes in this tree is the sum of the number of nodes in its children trees, plus one node for the root itself. Simplifying, we get

$$(2^{h+1}-1)+(2^{h+1}-1)+1 = 2(2^{h+1})-1 \tag{3}$$
$$= 2^{h+2}-1 \tag{4}$$
$$= 2^{(h+1)+1}-1 \tag{5}$$

A complete binary tree with height $h+1$ has $2^{(h+1)+1}-1$ total nodes. So by induction the hypothesis holds for all values of h.

□

2

Prove a complete binary tree with n nodes has $(n-1)/2$ internal nodes.

Proof. We will prove this statement directly.

Suppose you have a complete binary tree with n total nodes. Every complete binary tree has a number of nodes of the form 2^k-1, where k is an integer (this follows from #1 which was proved above). For this binary tree,

$$n = 2^k - 1 \tag{1}$$
$$n+1 = 2^k \tag{2}$$
$$\log_2(n+1) = k \tag{3}$$

We can add 1 to k to get the number of nodes in the next bigger complete binary tree.

$$2^{k+1} = 2^{\log_2(n+1)+1}-1 \tag{4}$$
$$= 2(2^{\log_2(n+1)})-1 \tag{5}$$
$$= 2(n+1)-1 \tag{6}$$

This bigger binary tree will have $2(n+1)-1 = 2n+1$ total nodes.
This means that we added

$$2n+1-n = n+1 \tag{7}$$

$n+1$ nodes to the new tree as leaves. So this bigger binary tree has $n+1$ leaves and $(2n+1)-(n+1) = n$ internal nodes.

Let a be the total number of nodes in this new tree.

$$a = n + 1 + n = 2n + 1 \tag{8}$$

It has n internal nodes. Writing n in terms of a, we get.

$$a = 2n + 1 \tag{9}$$
$$a - 1 = 2n \tag{10}$$
$$(a - 1)/2 = n \tag{11}$$

So a complete binary tree with a nodes has $(a - 1)/2$ internal nodes.

$\square$