Binary Trees (Induction/Recursion)

(Extended) binary trees are directed graphs with “nodes” and “links”, and are defined recursively as follows:

1. there are two kinds of nodes: NULL and non-NULL. Non-NULL nodes have two links, called left-child and right-child (they can also have other information, which we ignore).

2. one NULL node is a binary tree

3. If we have two disjoint binary tree $T_1$ and $T_2$, then a binary tree (which we call $T$) can be constructed with a root non-NULL node, whose left and right children are $T_1$ and $T_2$ respectively.

We define the height $h(T)$ of a binary tree $T$ to be the longest number of links one can traverse on a path in the tree. We define the size $s(T)$ of a binary tree $T$ to be the number of non-null nodes.

Then we have the recurrence relations:

$$s(T) = \begin{cases} 0 & \text{if } T == \text{NULL} \\ 1 + s(T_1) + s(T_2) & \text{if } T_1 == \text{left}_\text{child}(\text{root}(T)) \land T_2 == \text{right}_\text{child}(\text{root}(T)) \end{cases}$$

and

$$h(T) = \begin{cases} 0 & \text{if } T == \text{NULL} \\ 1 + \max\{h(T_1), h(T_2)\} & \text{if } T_1 == \text{left}_\text{child}(\text{root}(T)) \land T_2 == \text{right}_\text{child}(\text{root}(T)) \end{cases}$$

**Theorem 1** For any tree $T$, $s(T) \leq 2^{h(T)} - 1$. 