A Question of Some Depth

Let $T$ be a $k$-ary tree, i.e. each internal node of $T$ has at most $k$ children. Suppose that the depth of $T$ is $d$ (the maximum depth of any node of $T$ is $d$). What is the maximum number of *external nodes* (leaves) that $T$ can have? What is the maximum total number of nodes that $T$ can have?

**Solution:** The number of external nodes and the total number of nodes will clearly be maximized when every internal node has the maximum number of children.

For a $k$-ary tree, the root node then has $k$ children. Each of these children will have $k$ children of their own, so there will be $k^2$ children at depth 2 in $T$. Similarly, each of these $k^2$ nodes will have $k$ children, so there are $k^2 \cdot k = k^3$ vertices at depth 3, and so forth. The leaves of the tree will have depth $d$. Therefore, there will be $k^d$ nodes (leaves) at depth $d$. So the maximum number of leaves for a $k$-ary tree with depth $d$ is $k^d$.

Then the maximum total number of nodes is then

$$1 + k + k^2 + k^3 + \cdots + k^d = \frac{k^{d+1} - 1}{k - 1}.$$

**Note:** For your information, here we show that (assuming $k \neq 1$)

$$1 + k + k^2 + k^3 + \cdots + k^d = \frac{k^{d+1} - 1}{k - 1}.$$

(I don’t necessarily expect you to remember this formula, but more importantly, if you wanted to reproduce it, you should remember the idea behind you could get it.)

Let

$$S = k^d + k^{d-1} + \cdots + k^2 + k + 1.$$

Then

$$kS = k^{d+1} + k^d + \cdots + k^3 + k^2 + k.$$

Therefore

$$kS - S = (k^{d+1} + k^d + \cdots + k^2 + k) - (k^d + k^{d-1} + \cdots + k + 1)$$

and hence

$$(k - 1)S = k^{d+1} - 1.$$

So (dividing by $(k - 1)$ which is allowed since $k \neq 1$) we get that

$$S = \frac{k^{d+1} - 1}{k - 1}.$$
The Big Swap

Suppose that \( x \) and \( y \) are two integer variables. How can I swap the values of \( x \) and \( y \) without using a third temporary variable?

(The usual way to do this swap is to do something like

\[
\begin{align*}
\text{temp} & \leftarrow x \\
x & \leftarrow y \\
y & \leftarrow \text{temp}.
\end{align*}
\]

Here we want to avoid using this extra variable. What can we do instead to achieve this swap?)

Solution: I’ll just indicate how you can do this below. I will slightly abuse the notation and use \( x \) and \( y \) to denote the values of the variables. The idea is that in each step, we will perform one addition or subtraction, and in the last step, we will negate one number.

\[
\begin{align*}
x & \quad \Rightarrow \\
y & \quad \Rightarrow \\
x + y & \quad \Rightarrow \\
-x & \quad \Rightarrow \\
y & \quad \Rightarrow \\
x & \quad \Rightarrow
\end{align*}
\]

Note that, on a computer, this might run a chance of an integer overflow, i.e. if \( x \) and \( y \) are large integers, then computing the sum \( x + y \) might give a result that cannot be stored in an integer variable. (Of course, this is always true anytime an addition/subtraction operation is performed (or, indeed a multiplication or shift operation, etc.). The result might be too big to store in an integer variable in that programming language/operating system.)