Computing powers of a number

Although most languages have a builtin pow function that computes powers of a number, you can write a similar function recursively, and it can be very efficient. The only hitch is that the exponent has to be an integer.

Suppose you want to compute $\mathbf{x^n}$, where $\mathbf{x}$ is any real number and $\mathbf{n}$ is any integer. It's easy if $\mathbf{n}$ is $\mathbf{0}$, since $\mathbf{x^0} = \mathbf{1}$ no matter what $\mathbf{x}$ is. That's a good base case.

So now let's see what happens when $\mathbf{n}$ is positive. Let's start by recalling that when you multiply powers of $\mathbf{x}$, you add the exponents: $\mathbf{x^a} \cdot \mathbf{x^b} = \mathbf{x^{a+b}}$ for any base $\mathbf{x}$ and any exponents $\mathbf{a}$ and $\mathbf{b}$. Therefore, if $\mathbf{n}$ is positive and even, then $\mathbf{x^n} = \mathbf{x^{n/2}} \cdot \mathbf{x^{n/2}}$. If you were to compute $\mathbf{y} = \mathbf{x^{n/2}}$ recursively, then you could compute $\mathbf{x^n} = \mathbf{y} \cdot \mathbf{y}$. What if $\mathbf{n}$ is positive and odd? Then $\mathbf{x^n} = \mathbf{x^{n-1}} \cdot \mathbf{x}$, and $\mathbf{n-1}$ either is $\mathbf{0}$ or is positive and even. We just saw how to compute powers of $\mathbf{x}$ when the exponent either is 0 or is positive and even. Therefore, you could compute $\mathbf{x^{n-1}}$ recursively, and then use this result to compute $\mathbf{x^n} = \mathbf{x^{n-1}} \cdot \mathbf{x}$. What about when $\mathbf{n}$ is negative? Then $\mathbf{x^n} = \mathbf{1/x^{-n}}$, and the exponent $\mathbf{-n}$ is positive. So you can compute $\mathbf{x^{-n}}$ recursively and take its reciprocal. Putting these observations together, we get the following recursive algorithm for computing $\mathbf{x^n}$:

- The base case is when $\mathbf{n} = \mathbf{0}$, and $\mathbf{x^0} = \mathbf{1}$.
- If $\mathbf{n}$ is positive and even, recursively compute $\mathbf{y} = \mathbf{x^{n/2}}$, and then $\mathbf{x^n} = \mathbf{y} \cdot \mathbf{y}$. Notice that you can get away with making just one recursive call in this case, computing $\mathbf{x^{n/2}}$ just once, and then you multiply the result of this recursive call by itself.
- If $\mathbf{n}$ is positive and odd, recursively compute $\mathbf{x^{n-1}}$, so that the exponent either is $\mathbf{0}$ or is positive and even. Then, $\mathbf{x^n} = \mathbf{x^{n-1}} \cdot \mathbf{x}$
- If $\mathbf{n}$ is negative, recursively compute $\mathbf{x^{-n}}$, so that the exponent becomes positive. Then, $\mathbf{x^n} = \mathbf{1/x^{-n}}$.