

Re: Powers of 5

Source: <http://sci.tech-archive.net/Archive/sci.math/2006-09/msg00913.html>

- *From:* Rick Decker <rdecker@xxxxxxxxxxxxx>
 - *Date:* Tue, 05 Sep 2006 10:50:56 -0400
-

Josh wrote:

Rick Decker wrote:

<snip>

Powers of 5 of the form 2^k (i.e. $5^{(2^k)}$) can be computed quickly by successive squaring. For example,

$$5^{16} = (5^8)^2$$

$$5^8 = (5^4)^2$$

$$5^4 = (5^2)^2$$

$$5^2 = (5^1)^2$$

This should give you the modified algorithm almost immediately.

That's perfect. I wish I had thought of that. I rewrote the function in Python for the sake of being concise, but now the function takes k rather than 2^k as its argument. So I guess I haven't really answered the question.

```
def power(k):  
    if (k <= 1):  
        return 5  
    else:  
        return power(k-1)**2
```

<snip>

Also, I'm sort of curious what school is still using Pascal as an

Re: Powers of 5

introductory language. Not that I'm complaining—I think Pascal could still be a righteous choice for an intro vehicle; I'd just like to know what school still has the stones to resist jumping on the C#/C++/C/Java bandwagons.

That would be Drexel University. The course I've been taking covers data structures and algorithms, and therefore the programming component is small. Courses with greater emphasis on programming have entered into a sordid love affair with OO languages, especially C++ and Java. At least, that's how it seems to an outsider. (My home institution is the University of Chicago. They started us with Scheme, which made me loathe the nested parentheses for a while.)

At any rate, I digress. I'm just missing how to get the function to take 2^k as its argument. If you don't mind helping a little more, I would be grateful.

I'll do more than that. Here's the algorithm in a form that doesn't require the argument to be a power of 2. It's been a while since I programmed in Pascal, so bear with me.

```
function Power(x, n: integer) : integer;
{Returns  $x^n$  for  $n \geq 0$ .}
begin
  if n = 0 then
    Power := 1
  else
    if odd(n) then
      Power := x * Power(sqr(x), n div 2)
    else
      Power := Power(sqr(x), n div 2)
end;
```

This relies on the binary representation of n , so for example if $n = 13$ we have $x^{13} = x^{8+4+1} = x^8 * x^4 * x^1$ and the observation that if n is even then $x^n = (x^2)^{n/2}$ and if n is odd then $x^n = x * (x^2)^{\text{floor}(n/2)}$.

Trace the action of the function for $n = 13$ and you'll see what's going on.

Regards,

Rick

.