

5.4 Recursive algorithms: **Practice**

```
procedure fibonacci_rec (n: nonnegative integer)
if  $n = 0$  then return 0
else if  $n = 1$  then return 1
else return fibonacci_rec( $n-1$ ) + fibonacci_rec( $n-2$ )
{output:  $n^{\text{th}}$  Fibonacci number}
```

Show the figure for the call *fibonacci_rec*(5)

5.4 Recursive algorithms: **Practice**

Here is a recursive algorithm to compute r^n with some lines missing. The input r can be any real number. The input n is assumed to be a non-negative integer.

procedure *Exponent*(r : real number, n : non-negative integer)

----- // the base case

$p := \textit{Exponent}$ (-----) // the recursive call

return $r * p$

{output: r^n }

5.4 Recursive algorithms: **Practice**

Give recursive and interactive algorithms for finding the n^{th} term of the sequence defined by:

$$a_0 = 1, a_1 = 3, a_2 = 5, \text{ and } a_n = a_{n-1} a_{n-2}^2 a_{n-3}^3$$