

## Recursive algorithms: **Practice**

Consider another algorithm for finding the  $n^{\text{th}}$  Fibonacci number:

```
procedure fibonacci_rec2 (n: nonnegative integer)
if n = 0 then return 0
else if n = 1 then return 1
else return fibonacci_helper(1,1,0,n)
{output:  $n^{\text{th}}$  Fibonacci number}
```

```
procedure fibonacci_helper(curr,prev,i,n):
if i = n-2 then return curr
else return fibonacci_helper(curr+prev,curr,i+1,n)
```

Show the figure for the call *fibonacci\_rec2*(5)

## 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
```

```
 $p := Exponent$ ( ----- ) // the recursive call
```

```
return  $r * p$ 
```

```
{output:  $r^n$ }
```

## Recursive algorithms: **Practice**

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

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