Recursividad

Bach. Rodolfo Mercado Gonzales
Universidad Nacional de Ingeniería

Recursividad

- Enfoque poderoso que permite y facilita la solución de muchos problemas.
- Una función o procedimiento se denomina recursivo si se llama a sí mismo .

```
void recursiva ( … ) {
    …
    recursiva ( … )
     …
}
```

Recursividad

- Cada llamada a la función representa un nuevo problema.
- Presenta casos base, aquí no se usa recursión.
- No pueden existir ciclos en las llamadas.

Recursividad

- Existen funciones que ya tienen una definición explícitamente recursiva.

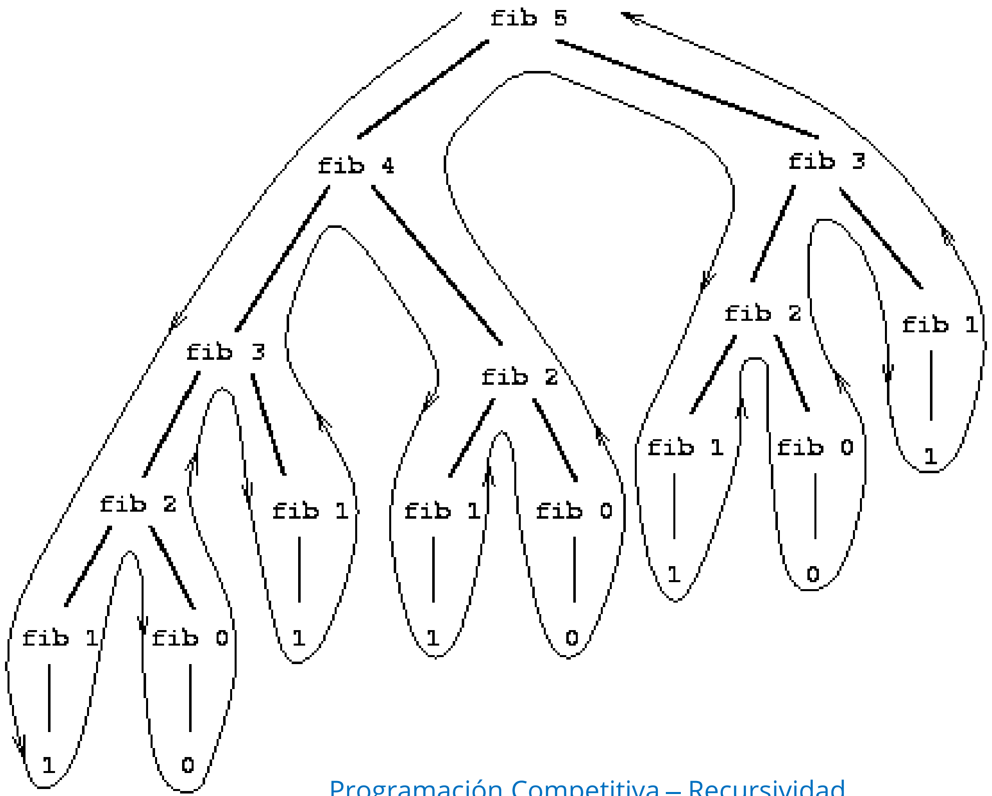
Sucesión de Fibonacci

$$F_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F_{n-1} + F_{n-2} & \text{if } n > 1 \end{cases}$$

Recursión Paso a Paso

```
                         fib(5)
                     /              \
             fib(4)                       fib(3)
           /        \                    /      \
      fib(3)          fib(2)          fib(2)     fib(1)
     /      \        /     \         /     \
 fib(2)    fib(1)  fib(1) fib(0)  fib(1) fib(0)
 /     \
fib(1) fib(0)
```

Recursión Paso a Paso



Recursividad

$$\gcd(a,b)\begin{cases} a, & b = 0 \\ \gcd(b, a\%b), & b > 0 \end{cases}$$

Máximo Común Divisor

Recursividad

- También podemos apoyarnos de una definición semántica para usar este enfoque.

Suma de los n primeros números naturales

Sea $\mathbf{S(n)}$ la suma de los números naturales desde 1 hasta n.

Podemos definir la siguiente recursión:

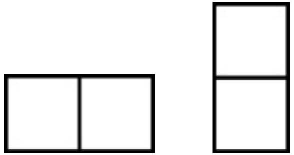
$$\mathbf{S(n) = S(n-1) + n} \qquad \mathbf{S}(1) = 1$$

Recursividad

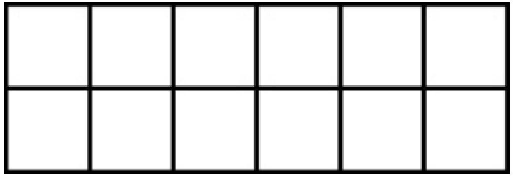
Dominós

¿De cuántas formas podemos cubrir un tablero de 2 x n usando dominós de 2 x 1 ?

Dominós de $2\ x\ 1$



Tablero de $2\ x\ n$



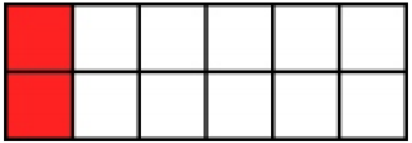
Recursividad

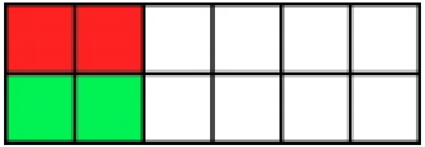
Sea $\boldsymbol{f(n)}$ el número de formas de cubrir un tablero de $2\ x\ n$.

Podemos definir la siguiente recursión:

$$\boldsymbol{f(n) = f(n-1) + f(n-2)}$$

$$f(0) = 1$$
$$f(1) = 1$$





Recursividad

Viaje Óptimo en una Matriz

Dado una matriz de $n\ x\ m$ con un número entero por casilla. Se desea obtener la máxima suma de números al recorrer las casillas desde la posición $(0,0)$ hasta $(n-1, m-1)$, únicamente haciendo movimientos hacia la derecha o abajo (se acumulan los valores de cada casilla visitada).

	0	1	2	3	4
0	5	9	-10	9	7
1	4	3	14	1	8
2	4	-14	-5	0	1
3	3	13	11	-3	3
4	-10	8	12	-9	4
5	1	4	-4	2	6

Recursión Paso a Paso

Sea $\boldsymbol{f(x,y)}$ la máxima suma que se puede obtener desde la casilla $(0{,}0)$ hasta (x,y).

Podemos definir las siguiente recursión:

	0	1	2	3	4
0	5	9	-10	9	7
1	4	3	14	1	8
2	4	-14	-5	0	1
3	3	13	11	-3	3
4	-10	8	12	-9	4
5	1	4	-4	2	6

$$f(x,y) = max\begin{cases} M_{xy} + f(x-1,y) \\ M_{xy} + f(x,y-1) \end{cases}$$

	0	1	2	3	4
0	5	9	-10	9	7
1	4	3	14	1	8
2	4	-14	-5	0	1
3	3	13	11	-3	3
4	-10	8	12	-9	4

	0	1	2	3
0	5	9	-10	9
1	4	3	14	1
2	4	-14	-5	0
3	3	13	11	-3
4	-10	8	12	-9
5	1	4	-4	2

Consideraciones

- Podemos realizar definiciones semánticas también para procedimientos.
- Si no se usa alguna técnica adicional la recursión tendrá complejidad exponencial.
- Debemos tener cuidado con la memoria stack (no almacenar más de 10^5 llamadas al mismo tiempo).

Aplicaciones

- Backtracking (fuerza bruta de forma recursiva)
- Divide y Vencerás
- Programación Dinámica

Problemas

UVA 10696 – f91

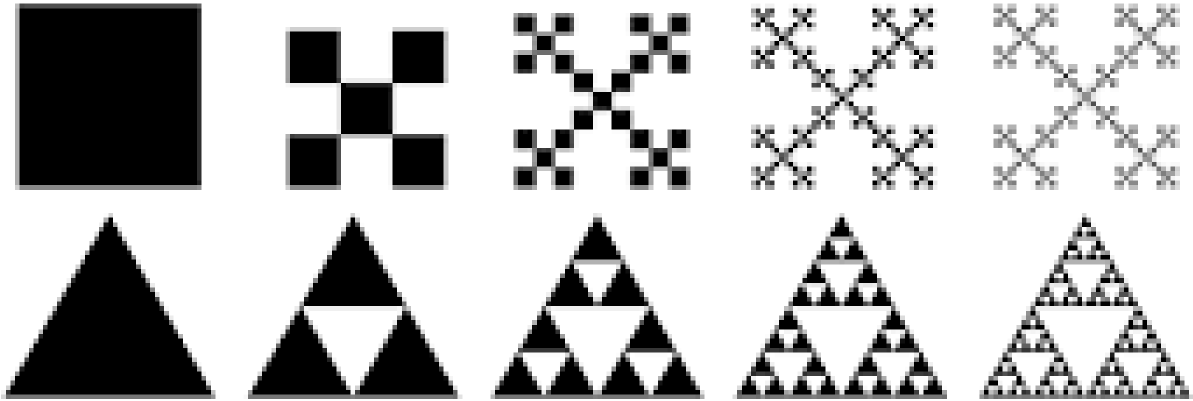
UVA 11332 – Summing Digits

UVA 10359 - Tiling

UVA 10017 – The Never Ending Towers of Hanoi

Fractales

Son figuras geométricas que se puede fraccionar en partes que son o se asemejan a una versión reducida de la figura completa.



The Sierpinski Fractal

$n = 1$

```
 /\
/__\
```

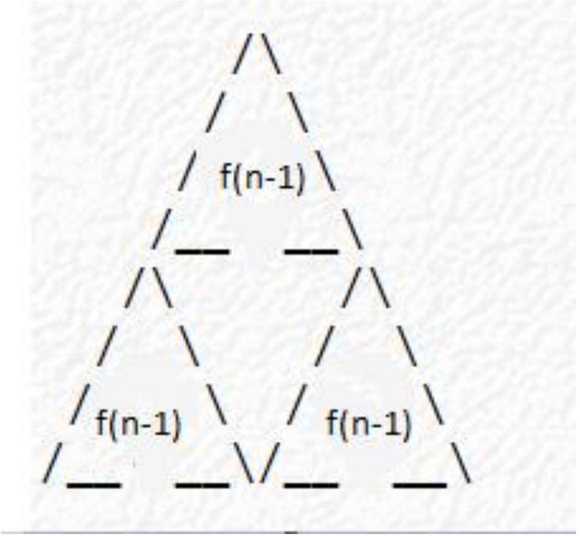
$n = 2$

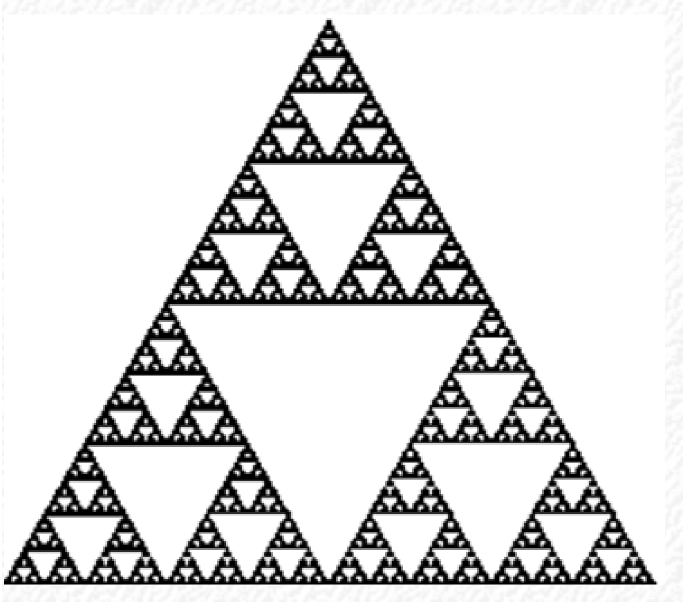
```
   /\
  /__\
 /\  /\
/__\/__\
```

$n = 3$

```
       /\
      /__\
     /\  /\
    /__\/__\
   /\      /\
  /__\    /__\
 /\  /\  /\  /\
/__\/__\/__\/__\
```

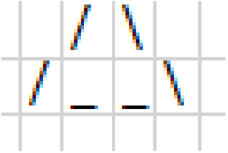
The Sierpinski Fractal

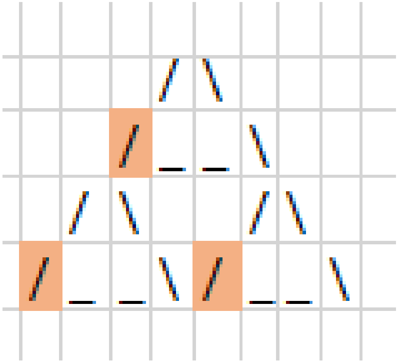




The Sierpinski Fractal

Solución:





```
dibujarFractal( int n, int x, y ){
    .....
    dibujarFractal( n-1, x, y );
    dibujarFractal( n-1, x - 2^(n-1), y + 2^(n-1) );
    dibujarFractal( n-1, x, y + 2^n );
    .....
}
```

Problemas

TJU 1406 – The Sierpinski Fractal

TJU 1178 – Fractal

Referencias

- Topcoder, An Introduction to Recursion

¡ Good luck and have fun !