

Práctico 4 – Recursividad

OBJETIVOS:

- Familiarizarse con el desarrollo de programas recursivos y comprender su funcionamiento.
- Comprender como se crean y destruyen los contextos de ejecución, para esto se recomienda realizar los diagramas de llamados.

Ejercicio 1

Implementar una función recursiva que calcule el factorial de un número natural N.

Ejercicio 2

a) Escriba una función que dado un número natural N, calcule el n-ésimo número de Fibonacci.

Los números de Fibonacci se obtienen de la siguiente manera.

$$\begin{aligned} fib(0) &= 0 \\ fib(1) &= 1 \\ fib(n) &= fib(n-1) + fib(n-2) \text{ si } n > 1 \end{aligned}$$

b) Realizar el árbol de llamadas para fib (n).

¿Es eficiente?

Realice el árbol de llamadas para fib(5).

¿Se le ocurre algún algoritmo que no calcule casos repetidos? En caso afirmativo impleméntelo.

Ejercicio 3

Diseñar una rutina que liste los números del 1 al n. ¿Podría lograr que los liste en orden inverso?

Ejercicio 4

Dado un arreglo de números enteros:

- Implementar una función que reciba el arreglo y retorne su mínimo elemento.
- Implementar una función que reciba el arreglo y retorne la posición del máximo elemento.
- Usando la función definida en la parte b) implemente una función recursiva que ordene un arreglo.

Ejercicio 5

Diseñar una rutina que dados dos números, a real y b entero, calcule la potencia a^b . La rutina debe funcionar correctamente para $b > 0$, $b = 0$ y $b < 0$.

Ejercicio 6

El máximo común divisor entre dos naturales x e y puede calcularse de la siguiente manera:

$$\begin{aligned} \text{mcd}(x,y) &= x, \text{ si } y=0 \\ \text{mcd}(x,y) &= \text{mcd}(y, x \bmod y) \end{aligned}$$

Dos números naturales son primos entre sí si su mcd es 1. Implemente un programa que dado dos números naturales, determine si son primos entre sí.

Ejercicio 7

Se considera una calle de adoquines. Sobre ella, se selecciona una línea de adoquines y se numera a partir de 0 de la siguiente forma:

0	1	2	...	N	N+1
---	---	---	-----	---	-----

Sobre esta hilera de adoquines hay niños jugando a la rayuela. Los movimientos permitidos del juego son:

- Ir de un adoquín numerado i al adoquín numerado $i+1$.
- Ir de un adoquín numerado i al adoquín $i+2$ (salteándose el adoquín $i+1$).
- Al principio del juego los niños se ubican en el adoquín numerado 0.

Se pide calcular el número de caminos posibles para alcanzar un adoquín objetivo numerado con n (mayor que cero).

Ejercicio 8

Considere la siguiente función:

```
static int x(int n, int k)
{
    int x1, x2;
    if (n < 1)
        return k;
    else {
        x1 = x(n-1, k);
        x2 = x(n-1, k);
        return (x1 + x2);
    }
}
```

¿Qué valor retorna $x(2,5)$? Justifique su respuesta realizando el árbol de llamadas.

Ejercicio 9

Considere la siguiente función:

```
static int v[] = {1,3,5,7,9,11,13,15};
static boolean b(int izq, int der, int x, int v[])
{
    if (izq > der)
        return false;
    else{
        int medio = (izq+der)/2;
        if (v[medio] == x)
            return true;
        else
            return (b(izq,medio-1,x,v) || b(medio+1,der,x,v));
    }
}
```

- a) Efectúe el diagrama de llamadas para b(0, 7, 13, v).
- b) ¿Qué retorna la función b?
- c) ¿Cómo podría mejorarla si se sabe que el vector v está ordenado?
- d) Escriba la versión mejorada de b.

Ejercicio 10

Considere el siguiente programa:

```
static int v[] = {2,5,3,7,9,1,8,40};

static int x(int v[], int izq, int der)
{
    if (izq > der)
        return 0;
    else{
        int medio = (izq+der)/2;
        return (v[medio]+x(v,izq,medio-1)+x(v,medio+1,der));
    }
}

public static void main(String []args)
{
    System.out.println("0 7 = " + x(v,0,7));
    System.out.println("2 2 = " + x(v,2,2));
    System.out.println("4 6 = " + x(v,4,6));
}
```

- a) ¿Qué valores muestra en pantalla?
- b) En general, ¿Cuál es el valor retornado para la función x?

Ejercicio 11

Desarrollar una función que reciba un mapa (representado por una matriz de MxN donde 0 representa agua y 1 tierra) y devuelva la cantidad de islas que contiene. Una isla es un conjunto de 1's adyacentes (vertical, horizontal u oblicuamente).

Por ejemplo en el siguiente mapa la respuesta sería 6 islas.

```
static int cuentoIslas(int mapa[][], int M, int N)
```

0	0	0	1	1	0	0	0	1	0
0	0	1	0	1	0	0	0	1	1
0	1	0	0	1	1	1	0	0	0
0	1	0	0	0	0	0	1	0	0
0	0	1	0	1	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1
0	0	0	1	1	1	0	0	1	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	1	1	0	0	1	0	1	0
0	0	0	0	0	0	0	0	0	1

Ejercicio 12

Desarrolle un algoritmo MinMax que recibe un vector, un entero desde y un entero hasta y retorna el mínimo y el máximo valor del vector entre las posiciones desde y hasta inclusive.