

5. Recursividad

5.1 Funciones Recursivas

Ejercicio 5.1.1

Calcular el factorial de un número positivo n . Tener en cuenta la definición matemática de $n!$:

$$n! = \begin{cases} 1, & \text{si } n = 0 \\ n \times (n - 1)!, & \text{si } n > 0 \end{cases}$$

Ejercicio 5.1.2

Dado un número n como parámetro de entrada, calcular el n -ésimo número de la serie de Fibonacci. Tener en cuenta la siguiente definición:

$$Fib(n) = \begin{cases} 1, & \text{si } n = 1, n = 2 \\ Fib(n - 1) + Fib(n - 2), & \text{si } n > 2 \end{cases}$$

Ejercicio 5.1.3

Dados dos números: a y b . Calcule la potencia a^b , usando sólo multiplicaciones sucesivas.

Ejercicio 5.1.4

Construir un algoritmo recursivo que permita determinar si los dígitos de un número n dado son todos pares.

Ejercicio 5.1.5

Dados dos números enteros, divídaluos (división entera) y muestre el resultado, usando sólo la operación resta.

Ejercicio 5.1.6

Determine recursivamente si un número dado es par o impar, usando sólo la operación resta.

Ejercicio 5.1.7

El algoritmo de Euclides para encontrar el MCD (máximo común divisor) de dos números enteros positivos (m y n) se puede definir recursivamente. Algoritmo de Euclides: el MCD de dos enteros es el entero mayor que divide a ambos.

Dividendo	Divisor	Cociente	Resto
m	n	q_1	r_1
n	r_1	q_2	r_2
r_1	r_2	q_3	r_3

Cuando el último resto es cero (por ej. $r_3 = 0$), el MCD es el último divisor (en ese caso, r_2). El algoritmo recursivo se puede definir con los siguientes pasos:

- $\text{MCD}(m, n) = n$, si $n \leq m$, n divide a m
- $\text{MCD}(m, n) = \text{MCD}(n, m)$, si $m < n$
- $\text{MCD}(m, n) = \text{MCD}(n, \text{resto de } m \text{ dividido por } n)$

Para simplificar el algoritmo considerar que siempre $m > n$

Ejercicio 5.1.8

Dado un vector de 10 números enteros, calcular la suma de sus elementos

5.2 Procedimientos Recursivas

Ejercicio 5.2.1

Imprimir las cifras de un número n (siendo $n \geq 0$) en orden inverso al original. Por ej.: el inverso de 254 es 452.

Ejercicio 5.2.2

Leer una palabra (una cadena de caracteres) y la cantidad de caracteres y generar su palíndromo. El palíndromo de "Venezuela" es "aleuzeneV".

Ejercicio 5.2.3

Dada una lista de nombres ordenada en forma ascendente, construir un procedimiento recursivo que imprima como salida la misma lista, pero en orden descendente, sin modificar la lista original.

Ejercicio 5.2.4

Para convertir un número decimal a binario, simplemente debe dividirse sucesivas veces por dos (2) hasta quedarnos con el cociente cero (0). Todos los restos de las divisiones, tomados en orden inverso, forman el número binario objetivo. Escribir un procedimiento recursivo que, recibiendo como parámetro un número entero positivo, muestre en pantalla el mismo número codificado en binario.

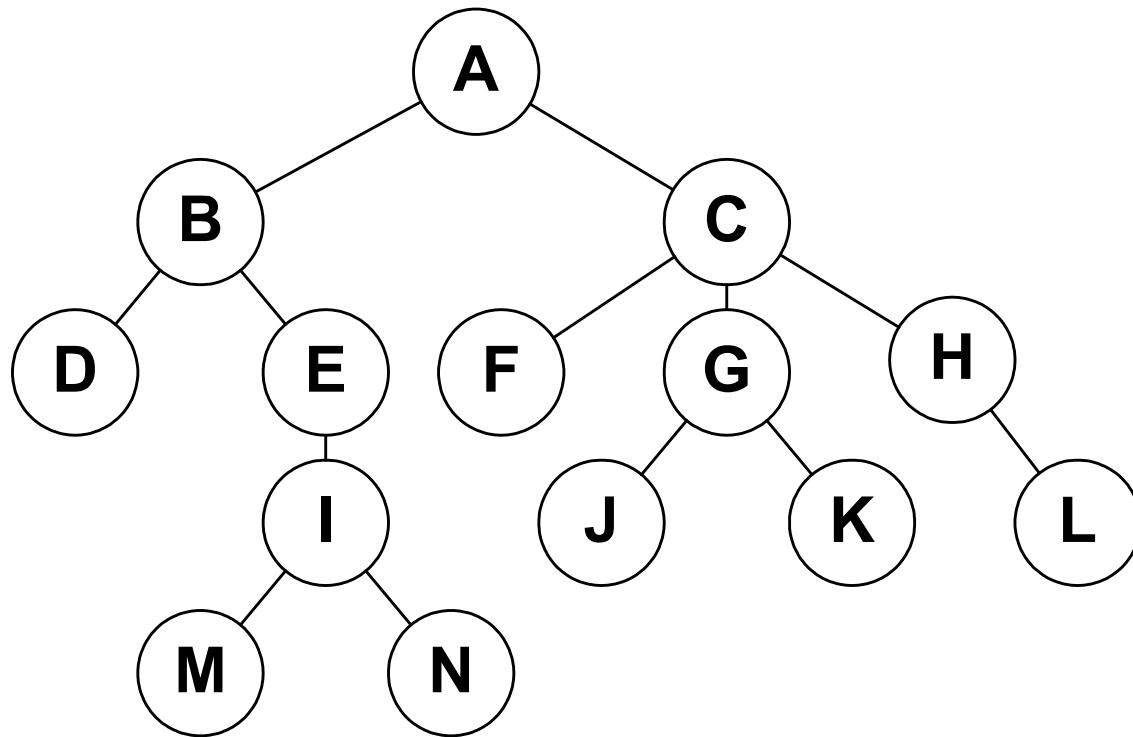
5.3 Arboles

Ejercicio 5.3.1

Responda las siguientes preguntas para el árbol de la figura:

1. ¿Qué nodos son hojas?

2. ¿Qué nodo es raíz?
3. ¿Cuál es el parent del nodo C?
4. ¿Qué nodos son los hijos de C?
5. ¿Qué nodos son los antecesores de C?
6. ¿Qué nodos son los descendientes de E?
7. ¿Cuáles son los hermanos derechos de D y E?
8. ¿Qué nodos están a izquierda y qué nodos a derecha de G?
9. ¿Cuál es la profundidad del árbol?
10. ¿Cuál es la altura del nodo C?



Ejercicio 5.3.2

¿Cuántos caminos de longitud 3 hay en el árbol representado en la figura anterior?

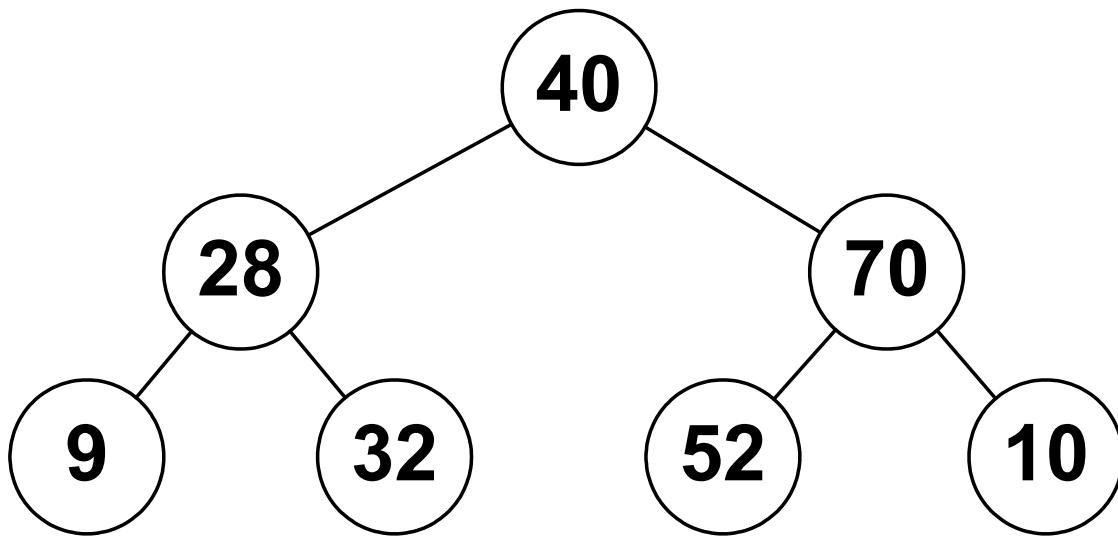
Ejercicio 5.3.3

Dada la expresión de la siguiente línea, dibujar el árbol equivalente

```
[ ( x - y ) * z ] / ( m + n ** p)
```

Ejercicio 5.3.4

Dado el siguiente árbol:



1. Indique el resultado del recorrido post-orden
2. Indique el resultado del recorrido en-orden
3. Elabore un algoritmo para el recorrido en-orden

Ejercicio 5.3.5

Para las siguientes expresiones:

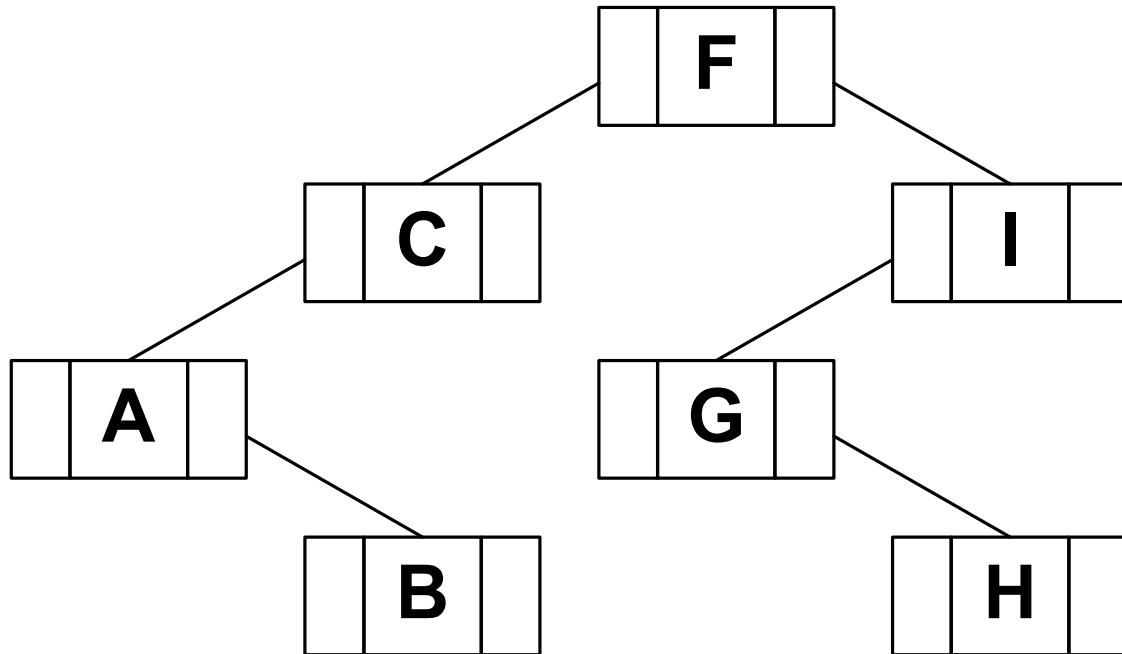
```
( 5 + 7 ) / 8 - ( 6 * 7 ) ** 2
3 - 6 + 6 * ( 8 / 3 )
4 / ( 8 - 6 * ( 8 + 3 ) )
```

1. Grafe el árbol equivalente
2. Indique las expresiones resultantes de los recorridos EN-ORDEN, PRE-ORDEN y POST-ORDEN

3. Calcule los resultados numéricos que arroja la ejecución de cada uno de los recorridos.

Ejercicio 5.3.6

Escriba un algoritmo que permita recorrer el siguiente árbol en los tres procedimientos, realice la prueba de escritorio



Ejercicio 5.3.7

Escribir una función recursiva que encuentre el número de nodos de un árbol binario.

Ejercicio 5.3.8

Escribir una función recursiva que encuentre la altura de un árbol binario.

Ejercicio 5.3.9

Se dispone de un árbol binario de enteros. Escribir funciones que calculen:

1. La suma de sus elementos.
2. La suma de sus elementos que son múltiplos de 3.

Ejercicio 5.3.10

Suponiendo que un árbol está definido como la estructura recursiva de datos:

```
arbol = registro
    x: entero
    izq, der: puntero a arbol
fin registro
```

Escribir un algoritmo que encuentre un elemento con una clave dada C, y realice una operación P con él.

Ejercicio 5.3.11

Conviértase la expresión $((a + b) + c * (d + e) + f) * (g + h)$ en

1. expresión prefija
2. expresión postfija

Ejercicio 5.3.12

El recorrido en pre-orden de un determinado árbol binario es: **GEAIBMCLDFKJH** y en-orden **IABEGLDCFMKHJ**.

1. Dibujar el árbol binario.
2. Dar el recorrido en post-orden.
3. Diseñar un algoritmo para recorrer el árbol en post-orden.

Ejercicio 5.3.13

Supongamos que tenemos una función valor tal que dado un valor de tipo char (una letra del alfabeto) devuelve un valor entero asociado a dicho identificador. Supongamos también la existencia de un árbol de expresión T cuyos nodos hoja son letras del alfabeto y cuyos nodos interiores son los caracteres *, +, -, /. Diseñar una función que tome como parámetros un nodo y un árbol binario y devuelva el resultado entero de la evaluación de la expresión representada.

Ejercicio 5.3.14

¿Puede reconstruirse de forma única un ABB dado su inorden? ¿Y dados el preorden y el postorden?

Ejercicio 5.3.15

Construir un ABB con las claves 50, 25, 75, 10, 40, 60, 90, 35, 45, 70, 42.

Ejercicio 5.3.16

Dados los siguientes árboles indicar cuál es AVL y cuál no.

