Trabajo Obligatorio

Recursividad

Leo Joaquin Bruno - FAI 3268

Maria Elvira Monserrat Vidal - FAI 1829

Jeremias Ezequiel Herrera - FAI 3297

29 de Mayo del 2022

Ejercicio 1 - Inciso D.

Realizar un proceso recursivo que devuelva el n-ésimo elemento de menor valor en una secuencia de enteros positivos. Por ejemplo, si la secuencia contiene los valores 12, 7, 3, 9, 1, 5, 21 y n = 1 devuelve 1, si n = 2 devuelve 3, si n = 5 devuelve 9 y si n = 28 devuelve -1, que indica que ha habido un error.

```
import java.util.Scanner;
public class resolucionEjercicioUno {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //Arreglo usado para testeo.
        int[] arreglo = {6,12,5};
        int n;
        System.out.println("Ingrese el valor de N");
        n = sc.nextInt();
        System.out.println(elemMenorValor(arreglo, n, 0));
        sc.close();
    }
    public static int elemMenorValor (int [] arreglo, int n, int posicion){
        //Dado un arreglo y un número, retorna el menor número al ingresado encontrado en el arreglo.
        int resultado;
        if(posicion!=arreglo.length-1){
            resultado = elemMenorValor(arreglo, n, posicion+1);
            if((arreglo[posicion]<resultado || resultado==-1) && arreglo[posicion]>=n){
                resultado = arreglo[posicion];
            }
        } else {
            if(arreglo[posicion]>=n){
                resultado = arreglo[posicion];
            } else {
                resultado = -1;
            }
        }
        return resultado;
    }
}
```

TRAZA.

MODULO elemMenorValor

arreglo	n	posicion	resultado	RETORNA
*	5	0		
			“” = ___ ➡ 5	5
		1		
			‘’ = ___ ➡ 5	5
		2		
			5	5

Arreglo: arreglo

6	12	5

Algoritmo principal.

arreglo	n	SALIDA
*		Ingrese el valor de N
	5	
		5

Ejercicio 2 - Inciso D.

Retornar verdadero si la matriz nxn es igual a su traspuesta.

JAVA.

```
public class resolucionEjercicioDos {
    public static void main(String[] args) {
        // Para probar matrices 3x3
        // int[][] matrizSimetrica = { { 1, 0, 4}, { 0, 2, 5 }, { 4, 5, 3 } };
        int[][] matrizAsimetrica = { { 1, 0, 4 }, { 0, 2, 8 }, { 4, 5, 3 } };

        // Para probar matrices 4x4
        //int[][] matrizSimetrica = { { 1, 0, 4, 7 }, { 0, 2, 5, 6 }, { 4, 5, 3, 0 }, { 7, 6, 0, 4 } };
        // int[][] matrizAsimetrica = { { 1, 0, 4, 7 }, { 0, 2, 5, 8 }, { 4, 5, 3, 0 }, { 7, 6, 0, 4 } };

        //System.out.println("Matriz: ");
        //leerArreglo2DRecursivo(matrizSimetrica, 0, 0);
        //System.out.print("\nLa matriz es igual a su transpuesta: ");
        //System.out.println(analizaSimetriaMatriz(matrizSimetrica, 0, 0));
        //System.out.println("-----------------------------------------------");
        System.out.println("Matriz: ");
        leerArreglo2DRecursivo(matrizAsimetrica, 0, 0);
        System.out.print("\nLa matriz es igual a su transpuesta: ");
        System.out.println(analizaSimetriaMatriz(matrizAsimetrica, 0, 0));
    }
    public static void leerArreglo2DRecursivo(int[][] arreglo, int i, int j) {
        // Imprime en pantalla el arreglo bidimensional de forma recursiva.
        System.out.print(arreglo[i][j] + " ");
        if (i != arreglo.length - 1 || j != arreglo[i].length - 1) {
            if (j == arreglo[i].length - 1) {
                i++;
                j = 0;
                System.out.println("");
            } else {
                j++;
            }
            leerArreglo2DRecursivo(arreglo, i, j);
        }
    }
```

```java
public static boolean analizaSimetriaMatriz(int[][] m, int i, int j) {
        //Revisa si la matriz es simétrica respecto a la diagonal principal. Retorna True de ser simétrica, false de no serlo.
        boolean existencia;
        if (compara(m[i][j], m[j][i])) {
            existencia = true;
            //Aumenta el valor de i o j.
            if (j == m[0].length - 1) {
                i++;
                j = 0;
            } else {
                j++;
            }
            //Evalúa que no se extienda del rango de la matriz.
            if (i != m.length - 1 || j != m[0].length - 1) {
                existencia = analizaSimetriaMatriz(m, i, j);
            }
        } else {
            //Si compara es falso, entonces existencia es falso.
            existencia = false;
        }
        return existencia;
    }
    public static boolean compara(int elemento1, int elemento2) {
        // Módulo que compara dos valores dados por parámetro, retorna true de ser iguales, false de ser distintos.
        boolean iguales = true;
        if (elemento1 != elemento2) {
            iguales = false;
        }
        return iguales;
    }
}
```

TRAZA.

Arreglo: matrizAsimetrica

1	0	4
0	2	8
4	5	3

Algoritmo principal

Salida
Matriz:
1 0 4 0 2 8 4 5 3
La matriz es igual a su transpuesta:
false

MODULO analizaSimetriaMatriz

m	i	j	existencia	RETORNA
1 0 4 0 2 8 4 5 3	0	0	analizaSimetriaMatriz(m, 0, 1) ➔ true	
	0	1	analizaSimetriaMatriz(m, 0, 2) ➔ true	
	0	2	analizaSimetriaMatriz(m, 1, 0) ➔ true	
	1	0	analizaSimetriaMatriz(m, 1, 1) ➔true	
	1	1	analizaSimetriaMatriz(m, 1, 2) ➔false	
	1	2		
				false

MODULO compara

elemento1	elemento2	RETORNA
1	1	true

elemento1	elemento2	RETORNA
0	0	true

elemento1	elemento2	RETORNA
4	4	true

elemento1	elemento2	RETORNA
0	0	true

elemento1	elemento2	RETORNA
2	2	true

elemento1	elemento2	RETORNA
8	5	false

Ejercicio 3 - Inciso A.

Retorna verdadero si un texto es palíndromo (no diferenciar mayúsculas de minúsculas). No se consideran mayúsculas o minúsculas. Ejemplo: Amor a roma.

JAVA.

```
import java.util.Scanner;
public class resolucionEjercicioTres {
    public static void main(String[] args) {
        // Verifica si una cadena es palindromo con un recorrido parcial
        Scanner sc = new Scanner(System.in);
        String frase;
        System.out.println("Ingrese una frase: ");
        frase = sc.nextLine();
        frase = frase.replaceAll(" ", "");
        frase = frase.toLowerCase();
        int der = frase.length() - 1;
        int izq = 0;
        System.out.println("¿Es palindromo?"+verificacionFrase(frase, izq, der));
        sc.close();
    }
    public static boolean verificacionFrase(String frase, int izq, int der) {
        //Verifica si una frase es palindromo.
        boolean similar = true;
        if (izq < der) { // Caso Recursivo: mientras la izq sea menor a la der
            if (frase.charAt(izq) == frase.charAt(der)) { // Se compara letra por letra de los extremos hacia el centro
                similar = verificacionFrase(frase, izq + 1, der - 1);
            } else {
                similar = false;  // Caso base: de no encontrar la similitud en al menos 1 letra retorna false
            }
        }
        return similar;
    }
}
```

TRAZA.

Algoritmo principal.

frase	izq	der	SALIDA
Amor a roma			
Amoraroma			
amoraroma	0		
		8	
			¿Es palindromo? true

MÓDULO verificacionFrase(amoraroma)

frase	izq	der	similar	RETORNA
amoraroma	0	8	verificacionFrase(frase,1,7) ➜true	
	1	7	verificacionFrase(frase,2,6) ➜true	
	2	6	verificacionFrase(frase,3,5) ➜true	
	3	5	verificacionFrase(frase,4,4) ➜true	
	4	4	true	
				true

Ejercicio 4 - Inciso A.

Dado un número decimal entero retorna su correspondiente número binario.

JAVA.

```
import java.util.Scanner;
public class resolucionEjercicioCuatro {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int numero;
        // número debe pertenecer a los enteros positivos
        do {
            System.out.println("Ingrese un número");
            numero = sc.nextInt();
            if (numero < 0) {
                System.out.println("El numero ingresado debe ser mayor o igual a 0");
            }
        } while (numero < 0);
        System.out.println(“El numero ”+numero+“ en binario es: ”+decimalABinario(numero));
        sc.close();
    }

    public static String decimalABinario(int num) {
        // Para convertir un numero decimal a binario:
        // -Tomamos los restos de las divisiones sucesivas de un número X
        // -Una vez hechas estas divisiones con sus respectivos restos, se anotan los
        // mismos de atrás hacia adelante
        String cadenaBinario = "";
        int restoNum;
        int cociente;

        if (num < 2) { // Mientras el numero sea menor a 2 se podría seguir dividiendo
            if (num != 0) {// Si el numero ingresado es negativo o 0, su valor será 0
                cadenaBinario = "1";// Cuando el número sea menor a 2 y distinto de 0, cadenaBinario se setea en 1 que es su último resto
            } else {
                cadenaBinario = "0";// Caso contrario el valor será 0
            }
        } else {
            restoNum = num % 2;
            // El resto de la división del número que se irá guardando en una cadena de
            // texto en el apilado
            cociente = num / 2;
            // Guardo el cociente del número para que entre en la recursividad
            cadenaBinario += decimalABinario(cociente) + restoNum;
        }
        return cadenaBinario;
    }
}
```

TRAZA.

Algoritmo principal

número	SALIDA
	Ingrese un número:
5	
	El número 5 en binario es: 101

MÓDULO decimalABinario(5) APILADO

num	restoNum	cociente	cadenaBinario	RETORNA
5	1	2	“” + decimalABinario(2) + 1	
2	0	1	“” + decimalABinario(1) + 0	
1			1	
				101

MÓDULO decimalABinario(5) DESAPILADO

num	restoNum	cociente	cadenaBinario	RETORNA
1			“1”	
2	0	1	“1” + 0	
5	1	2	“10”+1	
				101

Ejercicio 5 - Inciso D.

Una red de mensajes en cadena ha encontrado una forma de repartir los mismos de manera que se pueda cubrir a la mayor gente posible.....Para ello han decidido usar una teoría, la misma que empieza enviando X mensajes, y a partir de este número se van enviando los mensajes de acuerdo a una teoría progresiva. Se termina de entregar los mensajes una vez que se encuentra 1. Sin embargo, no se sabe a ciencia cierta la teoría, lo único que se tiene son algunas fórmulas sueltas: X/2 y X * 3 + 1 y ejemplos de cómo se aplica: CASO 1: 5, 16, 8, 4, 2, 1 CASO 2: 9, 28, 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2 , 1 CASO 3: 10, 5, 16, 8, 4, 2, 1 CASO 4: 6, 3, 10, 5, 16, 8, 4, 2, 1 CASO 5: 13, 40, 20, 10, 5, 16, 8, 4, 2, 1 a. En base a los casos deduce la teoría que se aplica para enviar los mensajes. b. Escribe el (los) método(s) RECURSIVO(S) necesario(s) para saber cuántos mensajes se han enviado en total si se empieza con X mensajes. Este resultado se obtiene sumando la cantidad de mensajes sucesivos que se envían. Para dar un ejemplo: en el CASO 1: se envían 36 mensajes.

JAVA.

```java
public class resolucionEjercicioCinco {
    public static void main(String[] args) {
        // Para 5 la secuencia es: 5, 16, 8, 4, 2, 1
        // La suma de mensajes debería ser: 36
        //int numero = 5;

        // Para 9 la secuencia es: 9, 28, 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2 , 1
        // La suma de mensajes debería ser: 339
        int numero = 5;
        System.out.println();
        mensajes(numero);
        System.out.println("\nLa suma de los valores da como resultado " + suma(numero) + " mensajes enviados.\n");
    }
    public static void mensajes(int num) {
        //Muestra la secuencia.
        System.out.print(num + " ");
        if (num != 1) { //Caso Base: Mientras el número sea distinto de 1, seguirá entrando en recursividad
            if (num % 2 == 0) { //Si el número es divisible por 2, entra en recursividad con el número dividido 2
                mensajes(num / 2);
            } else { //Caso contrario, el número se multiplica por 3 y se le suma 1
                mensajes((num * 3) + 1);
            }
        }
    }
    public static int suma(int num) {
        //Realiza la suma recursiva de los valores de la secuencia a partir del número inicial.
        int valores;
        if (num == 1) { //Caso base. A partir de este valor, comenzará la suma de los valores apilados.
            valores = 1;
        } else {  //De no ser el caso base, sigue entrando en recursividad.
            if (num % 2 == 0) { //Si el número es divisible por 2, entra en recursividad con el número dividido 2
                valores = num + suma(num / 2);
            } else { //Caso contrario, el número se multiplica por 3 y se le suma 1
                valores = num + suma((num * 3) + 1);
            }
        }
        return valores;
    }
}
```

TRAZA.

TRAZA MODULO mensajes(5)

num	SALIDA
5	5
mensajes((5*3)+1)	16
mensajes(16/2)	8
mensajes(4/2)	4
mensajes(4/2)	2
mensajes(2/2)	1

TRAZA MÓDULO suma(5)

num	valores	RETORNA
5		36
	5 + __ ➔ 31 + 5 = 36	
16		
	16 +__ ➔ 16 + 15 = 31	
8		
	8 +__ ➔ 8 + 7 = 15	
4		
	4 +__ ➔ 4 + 3 = 7	
2		
	2 +__ ➔ 2 + 1 = 3	
1		