Práctico 10 – Recursividad (parte 1)

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)

Ejercicios previos

Antes de comenzar el practico sugerimos analizar en forma recursiva e iterativa los siguientes métodos:

- public int potencia(int a, int b);
- public void mostrarasc(int n); //Muestra los números desde 1 hasta n
- public void mostrardsc(int n); //Muestra los números de n hasta 1

Ejercicio 1

Implementar una función recursiva que calcule el factorial de un número natural N.
Ejemplo: 5! = 5*4*3*2*1 = 120
Ver definición

Firma a utilizar:

- public int factorial(int n);

Ejercicio 2

Implementar un algoritmo que dado un número natural N, calcule el n-ésimo número de Fibonacci.
Ver definición

Los números de Fibonacci se obtienen de la siguiente manera.

fib(0) = 0
fib(1) = 1
fib(n)=fib(n-1)+fib(n-2) si n>1

Firma a utilizar:

- public int fib(int n);

a) 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

Implementar un algoritmo recursivo que permita invertir una palabra.
Ejemplo: Entrada: Hola Salida: aloH

Firma a utilizar:
public String invertir(String palabra);

a) Implmentarlo con la firma dada
b) Analizar como se podría implementar agregando un nuevo parámetro: int indice

Ejercicio 4

Implementar un algoritmo recursivo que permita sumar los dígitos de un número.
Ejemplo: Entrada: 123 Resultado: 6
Firma a utilizar:
public int sumar(int n);

Ejercicio 5

Implementar un algoritmo recursivo que permita sumar los elementos de un vector. Dibuje el diagrama de llamadas para el vector[] = {34,2,11,2}
Firma a utilizar:
public int sumavector(int []v);

Ejercicio 6

Implementar un algoritmo recursivo que permita multiplicar los elementos de un vector
Firma a utilizar:
public int multiplicarVector(int[] vec);

Ejercicio 7

Dado un arreglo de números enteros:

a) Implementar una función recursiva que calcule el máximo valor del vector. Dibuje el diagrama de llamadas para el vector [] = {34,11,35,1}
b) Implementar una función recursiva que calcule el mínimo valor del vector.

Ejercicio 8

Dado un arreglo de números enteros, implementar una función recursiva que indique si se encuentra determinado número en un vector.

a) Implementar una solución para un vector desordenado.
b) Implementar una solución para un vector ordenado

Ejercicio 9

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 árblo de llamadas.

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?

Ejercicios Complementarios.

Ejercicio complementario 1

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.

Ejercicio complementario 2

Implementar un algoritmo recursivo que permita hacer la división por restas sucesivas.
Firma a utilizar:
public int divide(int a, int b);

Ejercicio complementario 3

Implementar un algoritmo recursivo que permita hacer una multiplicación, utilizando el método Ruso.
Para más información: aquí.

Firma a utilizar:
public int multiplicarusa(int a, int b);