Metodología de la Programación
Grado en Ingeniería Informática
Guión de Prácticas

RECURSIVIDAD

Objetivos

- Aprender a resolver problemas de forma recursiva.
- Dominar los métodos de transformación de algoritmos recursivos.

En las prácticas de este tema, el alumno deberá:

- Diseñar un subalgoritmo recursivo para todos los problemas del bloque I que se presentan a continuación.
- Implementar en C los subalgoritmos recursivos del bloque I.

- Obtener, para todos los problemas del bloque II, una función recursiva final equivalente a la función recursiva que se presenta y las correspondientes soluciones iterativas, detallando todos los pasos en cada una de las transformaciones, siguiendo los métodos de transformación explicados en la teoría de la asignatura.
- Implementar en C las versiones recursivas e iterativas del paso anterior.

BLOQUE I - Implementación de subalgoritmos recursivos

1.- Diseñe una función recursiva que calcule el Máximo Común Divisor de dos números *a* y *b* utilizando el algoritmo de Euclides.

2.- Si *conj(n,k)* representa la cantidad de diferentes conjuntos de *k* personas que pueden formarse, dadas *n* personas entre las cuales elegir. Por ejemplo, *conj*(4,3) = 4, porque dadas cuatro personas A, B, C y D hay cuatro conjuntos posibles de tres personas: ABC, ABD, ACD y BCD. En general se cumple la siguiente relación:
conj(n,k) = conj(n-1,k) + conj(n-1,k-1)

Diseñe una función recursiva para calcular *conj(n,k)* para $n,k \geq 1$.

3.- Diseñe una función recursiva que calcule el producto de dos números mediante la multiplicación rusa.

4.- Dado un vector *n* de enteros, se dice que un elemento del vector es elemento mayoritario si este entero aparece estrictamente más de *n*/2 veces en dicho vector. Diseñe un algoritmo que determine de forma recursiva si el vector A[1..n] contiene un elemento mayoritario y la primera posición que ocupa.

5.- Diseñe una función recursiva que calcule la función de Ackermann para dos valores cualesquiera *m* y *n*.

6.- Dados dos vectores A y B de *n* y *m* elementos enteros, respectivamente, cumpliéndose que *n*>=*m*, que ningún elemento se repite y que están ordenados crecientemente, diseñe una función recursiva que determine si todos los elementos de B están contenidos en A.

7.- Diseñe un algoritmo que calcule de forma recursiva la suma de todos los elementos *i* de un vector A de *n* enteros que cumplen la siguiente propiedad:

$$1 \le i \le \left(\frac{n}{2}\right) - 1 : A[i] > A[2*i] \wedge A[i] > A[2*i+1]$$

8.- En un vector de enteros se genera un «cambio de tendencia» cuando dada una secuencia creciente o decreciente de números que ocupan posiciones consecutivas del vector, el elemento que le sucede es inferior o superior, respectivamente. Dado un vector de N enteros, diseñe una función recursiva que calcule el número de «cambios de tendencia» que contiene dicho vector.

9.- Dado un conjunto de N puntos en el plano, diseñe una función recursiva que encuentre el par de puntos con distancia mínima y devuelva dicho valor.

Nota.- Puede basarse en el teorema de Pitagóras.

BLOQUE II – Transformación de subalgoritmos recursivos

Se supone la existencia del tipo *Vect* definido como: **vector**[N] de entero: Vect

10.-

```
entero función fun(E Vect: x, E entero: n E entero: i)
 { x = A[1..n] ∧ n≥0  ∧ 0≤ i≤n }
 inicio
     si i=0 entonces
        devolver 0
     si_no
        devolver x[i] + fun(x, n, i-1)
     fin_si
 {devuelve la suma de los i primeros elementos del vector}
 fin_función
```

11.- entero **funcion** fun (E Vect: x, E Vect: y, E Vect: z, E entero: n, E entero: i)
{x = A[1..n] ∧ y = B[1..n] ∧ z=C[1..n] ∧ 1≤ i≤n }
inicio
si i=n **entonces**
devolver x[i]*y[i] + y[i]*z[i]
si_no
devolver x[i]*y[i] + y[i]*z[i] + 5*fun(x, y , z, n, i+1)
fin_si
{devuelve $\sum_{\alpha=i}^{n}(x[\alpha]*y[\alpha]+y[\alpha]*z[\alpha])*5^{\alpha-i}$}
fin_función

12.- entero **funcion** fun (E Vect: x, E Vect: y, E entero: n, E entero: i)
{x = A[1..n] ∧ y = B[1..n] ∧ 1≤i ∧ i≤n}
inicio
si i>n **entonces**
devolver 1
si_no
devolver (6*x[i] + 6*y[i]) * fun(x, y, n, i+1)
fin_si
{devuelve $\prod_{\alpha=i}^{n}6*(x[\alpha]+y[\alpha])$}
fin_función

13.- real **funcion** fun (E Vect: x, E Vect: y, E entero: n, E entero: i)
{x = A[1..n] ∧ y = B[1..n] ∧ 1≤ i≤n }
inicio
si i=n **entonces**
devolver 3*x[i]*y[i]
si_no
devolver 3*x[i]*y[i] + □*fun(x, y, n, i+1)
fin_si
{devuelve $\sum_{\alpha=i}^{n}\frac{i!}{\alpha!}(3*x[\alpha]*y[\alpha])$}
fin_función