ALGORITMIA

Tema 2:

Diseño de Algoritmos Recursivos

Solución Ejercicios: Inmersión no final

Curso 2021-2022

Grado en Ingeniería Informática en Tecnologías de la Información
Escuela Politécnica de Ingeniería – Campus de Gijón
Universidad de Oviedo

Ejercicio 1

Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

$Q \equiv \{ n \geq 1 \wedge x \geq 0 \}$
Funcion F1(A[1..n]:vector de enteros, x:entero) retorna (p:entero)
$R \equiv \{ p = (\Pi\, i)\, (A[i] + x^i : 1 \leq i \leq n) \}$

SOLUCIÓN 1.- Para obtener R´ partir de R, vamos a sustituir la constante **n por j**, el algoritmo resultante es el siguiente:

$Q' \equiv \{ 1 \leq j \leq n \wedge x \geq 0 \}$

Función iF1(A[1..n]:vector de enteros, x:entero, j:entero) retorna (p:entero)
caso
$j = 1 \rightarrow A[j] + x^j$
$j > 1 \rightarrow iF1(A, x, j-1) * (A[j] + x^j)$
fcaso
ffuncion
$R' \equiv \{ p = (\Pi\, i)\, (A[i] + x^i : 1 \leq i \leq j) \}$

La llamada inicial a la función se realizará con **j = n**, de ese modo.-

F1(A, x) = iF1(A, x, n)

SOLUCIÓN 2.- Para obtener R´ partir de R, vamos a sustituir la constante **1 por j**, el algoritmo resultante es el siguiente:

$Q' \equiv \{ 1 \leq j \leq n \wedge x \geq 0 \}$

Función iF1(A[1..n]:vector de enteros, x:entero, j:entero) retorna (p:entero)
caso
$j = n \rightarrow A[j] + x^j$
$j < n \rightarrow iF1(A, x, j+1) * (A[j] + x^j)$
fcaso
ffuncion
$R' \equiv \{ p = (\Pi\, i)\, (A[i] + x^i : j \leq i \leq n) \}$

La llamada inicial a la función se realizará con **j = 1**, de ese modo.-

F1(A, x) = iF1(A, x, 1)

Ejercicio 2

Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

$Q \equiv \{ n \geq 1 \wedge x \geq 0 \}$
Funcion F2(A[1..n]:vector de enteros, x:entero) retorna (p:booleano)
$R \equiv \{ p = (\exists i)(A[i] = x^i : 1 \leq i \leq n) \}$

SOLUCIÓN 1.- Para obtener R´ partir de R, vamos a sustituir la constante **n por j**, el algoritmo resultante es el siguiente:

$Q' \equiv \{ 1 \leq j \leq n \wedge x \geq 0 \}$
Función iF2(A[1..n]:vector de enteros, x:entero, j:entero) retorna (p:booleano)

```
caso
    j = 1  →  ( A[ j ] = x^j )
    j > 1  →  iF2(A, x, j-1 ) v (A[ j ] = x^j )
fcaso
ffuncion
```

$R' \equiv \{ p = (\exists i)(A[i] = x^i : 1 \leq i \leq j) \}$

La llamada inicial a la función se realizará con **j = n**, de ese modo.-

F2(A,x) = iF2(A, x, n)

SOLUCIÓN 2.- Para obtener R´ partir de R, vamos a sustituir la constante **1 por j**, el algoritmo resultante es el siguiente:

$Q' \equiv \{ 1 \leq j \leq n \wedge x \geq 0 \}$
Función iF2(A[1..n]:vector de enteros, x:entero, j:entero) retorna (p:booleano)

```
caso
    j = n  →  ( A[ j ] = x^j )
    j < n  →  iF2(A, x, j+1 ) v (A[ j ] = x^j )
fcaso
ffuncion
```

$R' \equiv \{ p = (\exists i)(A[i] = x^i : j \leq i \leq n) \}$

La llamada inicial a la función se realizará con **j = 1**, de ese modo.-

F2(A,x) = iF2(A, x, 1)

Ejercicio 3

Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

$Q \equiv \{ n \geq 1 \}$
Funcion F3(A[1..n]:vector de enteros) retorna (p : entero)
$R \equiv \{ p = (N\ i) (A[i] = 3 : 1 \leq i \leq n) \}$

SOLUCIÓN 1.- Para obtener R´ partir de R, vamos a sustituir la constante **n por j**, el algoritmo resultante es el siguiente:

$Q' \equiv \{ 1 \leq j \leq n \}$
Función iF3(A[1..n]:vector de enteros, j:entero) retorna (p:entero)

```
        caso
                j = 1     →  ( A[ j ] = 3 )
                j > 1     →  iF3(A, j-1 )  + ( A[ j ] = 3 )
        fcaso
ffuncion
```

$R' \equiv \{ p = (N\ i) (A[i] = 3 : 1 \leq i \leq j) \}$

donde (A[j] = 3) tendrá valor 1 si es cierto y 0 si es falso.

La llamada inicial a la función se realizará con **j = n**, de ese modo.-

F3(A) = iF3(A, n)

SOLUCIÓN 2.- Para obtener R´ partir de R, vamos a sustituir la constante **1 por j**, el algoritmo resultante es el siguiente:

$Q' \equiv \{ 1 \leq j \leq n \}$
Función iF3(A[1..n]:vector de enteros, j:entero) retorna (p:entero)

```
        caso
                j = n     →  ( A[ j ] = 3 )
                j < n     →  iF3(A, j+1 )  + ( A[ j ] = 3 )
        fcaso
ffuncion
```

$R' \equiv \{ p = (N\ i) (A[i] = 3 : j \leq i \leq n) \}$

donde (A[j] = 3) tendrá valor 1 si es cierto y 0 si es falso.

La llamada inicial a la función se realizará con **j = 1**, de ese modo.-

F3(A) = iF3(A, 1)

Ejercicio 4

Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

$Q \equiv \{ n \geq 1 \}$
Funcion F4(A[1..n]:vector de enteros) retorna (p : entero)
$R \equiv \{ p = (\text{MAX } i)\ (A[i] : 1 \leq i \leq n) \}$

SOLUCIÓN 1.- Para obtener R´ partir de R, vamos a sustituir la constante **n por j**, el algoritmo resultante es el siguiente:

$Q' \equiv \{ 1 \leq j \leq n \}$
Función iF (A[1..n]:vector de enteros, j:entero) retorna (p :entero)

```
    caso
        j = 1    → A[ j ]
        j > 1    → maximo ( iF4(A, j-1) , A[ j ] )
    fcaso
ffuncion
```

$R' \equiv \{ p = (\text{MAX } i)\ (A[i] : 1 \leq i \leq j) \}$

donde la función maximo(a,b) = a si $a \geq b$ y maximo(a,b) = b si $a < b$.

La llamada inicial a la función se realizará con **j = n**, de ese modo.-

F4(A) = iF4(A, n)

SOLUCIÓN 2.- Para obtener R´ partir de R, vamos a sustituir la constante **1 por j**, el algoritmo resultante es el siguiente:

$Q' \equiv \{ 1 \leq j \leq n \}$
Función iF4(A[1..n]:vector de enteros, j:entero) retorna (p :entero)

```
    caso
        j = n    → A[ j ]
        j < n    → maximo( iF4(A, j+1) , A[ j ] )
    fcaso
ffuncion
```

$R' \equiv \{ p = (\text{MAX } i)\ (A[i] : j \leq i \leq n) \}$

donde la función maximo(a,b) = a si $a \geq b$ y maximo(a,b) = b si $a < b$.

La llamada inicial a la función se realizará con **j = 1**, de ese modo.-

F4(A) = iF4(A, 1)

Ejercicio 5

Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

$Q \equiv \{ n \geq 0 \wedge x \geq 0 \}$
Funcion F5(A[1..n]:vector de enteros, x:entero) retorna (p:entero)
$R \equiv \{ p = (\Pi\, i)\, (A[i] + x^i : 1 \leq i \leq n) \}$

SOLUCIÓN 1.- Para obtener R´ partir de R, vamos a sustituir la constante **n por j**, el algoritmo resultante es el siguiente.-

$Q' \equiv \{ 0 \leq j \leq n \wedge x \geq 0 \}$

Función iF5(A[1..n]:vector de enteros, x:entero, j:entero) retorna (p:entero)
caso
$j = 0 \rightarrow 1$
$j > 0 \rightarrow$ iF5(A, x, j-1) * ($A[j] + x^j$)
fcaso
ffuncion
$R' \equiv \{ p = (\Pi\, i)\, (A[i] + x^i : 1 \leq i \leq j) \}$

La llamada inicial a la función se realizará con **j = n**, de ese modo.-

F5(A, x) = iF5(A, x, n)

SOLUCIÓN 2.- Para obtener R´ partir de R, vamos a sustituir la constante **1 por j**, el algoritmo resultante es el siguiente.-

$Q' \equiv \{ 1 \leq j \leq n+1 \wedge x \geq 0 \}$

Función iF5(A[1..n]:vector de enteros, x:entero, j:entero) retorna (p:entero)
caso
$j = n+1 \rightarrow 1$
$j < n+1 \rightarrow$ iF5(A, x, j+1) * ($A[j] + x^j$)
fcaso
ffuncion
$R' \equiv \{ p = (\Pi\, i)\, (A[i] + x^i : j \leq i \leq n) \}$

La llamada inicial a la función se realizará con **j = 1**, de ese modo.-

F5(A, x) = iF5(A, x, 1)

Ejercicio 6

Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

$Q \equiv \{ n \geq 0 \wedge x \geq 0 \}$
Funcion F6(A[1..n]:vector de enteros, x:entero) retorna (p:booleano)
$R \equiv \{ p = (\exists i)(A[i] = x^i : 1 \leq i \leq n) \}$

SOLUCIÓN 1.- Para obtener R´ partir de R, vamos a sustituir la constante **n por j**, el algoritmo resultante es el siguiente.-

$Q' \equiv \{ 0 \leq j \leq n \wedge x \geq 0 \}$
Función iF6(A[1..n]:vector de enteros, x:entero, j:entero) retorna (p:booleano)
caso
$j = 0 \rightarrow$ FALSO
$j > 0 \rightarrow$ iF6(A, x, j-1) v $(A[j] = x^j)$
fcaso
ffuncion
$R' \equiv \{ p = (\exists i)(A[i] = x^i : 1 \leq i \leq j) \}$

La llamada inicial a la función se realizará con **j = n**, de ese modo.-

F6(A,x) = iF6(A, x, n)

SOLUCIÓN 2.- Para obtener R´ partir de R, vamos a sustituir la constante **1 por j**, el algoritmo resultante es el siguiente.-

$Q' \equiv \{ 1 \leq j \leq n+1 \wedge x \geq 0 \}$
Función iF6(A[1..n]:vector de enteros, x:entero, j:entero) retorna (p:booleano)
caso
$j = n+1 \rightarrow$ FALSO
$j < n+1 \rightarrow$ iF6(A, x, j+1) v $(A[j] = x^j)$
fcaso
ffuncion
$R' \equiv \{ p = (\exists i)(A[i] = x^i : j \leq i \leq n) \}$

La llamada inicial a la función se realizará con **j = 1**, de ese modo.-

F6(A,x) = iF6(A, x, 1)

Ejercicio 7

Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

$Q \equiv \{ n \geq 0 \}$
Funcion F7(A[1..n]:vector de enteros) retorna (p : entero)
$R \equiv \{ p = (N\ i) (A[i] = 3 : 1 \leq i \leq n) \}$

SOLUCIÓN 1.- Para obtener R´ partir de R, vamos a sustituir la constante **n por j**, el algoritmo resultante es el siguiente.-

$Q' \equiv \{ 0 \leq j \leq n \}$
Función iF7(A[1..n]:vector de enteros, j:entero) retorna (p:entero)

```
        caso
                j = 0    → 0
                j > 0    → iF7(A, j-1 )  + ( A[ j ] = 3 )
        fcaso
ffuncion
```

$R' \equiv \{ p = (N\ i) (A[i] = 3 : 1 \leq i \leq j) \}$

donde (A[j] = 3) tendrá valor 1 si es cierto y 0 si es falso.

La llamada inicial a la función se realizará con **j = n**, de ese modo.-

F7(A) = iF7(A, n)

SOLUCIÓN 2.- Para obtener R´ partir de R, vamos a sustituir la constante **1 por j**, el algoritmo resultante es el siguiente.-

$Q' \equiv \{ 1 \leq j \leq n+1 \}$
Función iF7(A[1..n]:vector de enteros, j:entero) retorna (p:entero)

```
        caso
                j = n+1    → 0
                j < n+1    → iF7(A, j+1 )  + ( A[ j ] = 3 )
        fcaso
ffuncion
```

$R' \equiv \{ p = (N\ i) (A[i] = 3 : j \leq i \leq n) \}$

donde (A[j] = 3) tendrá valor 1 si es cierto y 0 si es falso.

La llamada inicial a la función se realizará con **j = 1**, de ese modo.-

F7(A) = iF7(A, 1)

Ejercicio 8

Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

Q ≡{ n ≥ 0 }
Funcion F8(A[1..n]:vector de enteros) retorna (p : entero)
R ≡{ p = (MAX i) (A[i] : 1≤ i ≤ n) }

SOLUCIÓN 1.- Para obtener R´ partir de R, vamos a sustituir la constante **n por j**, el algoritmo resultante es el siguiente.-

Q´ ≡ { 0 ≤ j ≤ n }
Función iF8(A[1..n]:vector de enteros, j:entero) retorna (p :entero)
caso
j = 0 → - ∞
j > 0 → maximo (iF8(A, j-1) , A[j])
fcaso
ffuncion
R´ ≡ { p = (MAX i) (A[i] : 1≤ i ≤ j) }

donde la función maximo(a,b) = a si a ≥ b y maximo(a,b) = b si a < b.

La llamada inicial a la función se realizará con **j = n**, de ese modo.-

F8(A) = iF8(A, n)

SOLUCIÓN 2.- Para obtener R´ partir de R, vamos a sustituir la constante **1 por j**, el algoritmo resultante es el siguiente.-

Q´ ≡ { 1 ≤ j ≤ n+1 }
Función iF8(A[1..n]:vector de enteros, j:entero) retorna (p :entero)
caso
j = n+1 → - ∞
j < n+1 → maximo (iF8(A, j+1) , A[j])
fcaso
ffuncion
R´ ≡ { p = (MAX i) (A[i] : j ≤ i ≤ n) }

donde la función maximo(a,b) = a si a ≥ b y maximo(a,b) = b si a < b.

La llamada inicial a la función se realizará con **j = 1**, de ese modo.-

F8(A) = iF8(A, 1)