ALGORITMIA

Tema 2:

Diseño de Algoritmos Recursivos

Solución de Problemas de Exámenes de cursos anteriores

Curso Académico 2021-2022

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

Curso Académico 2020-2021

1.- [7 puntos] Se quiere diseñar una función recursiva que resuelva el siguiente problema: "Dada una matriz M[1..n][1..m] siendo n y m mayores que 0, dicha función tiene que comprobar si todos los elementos cumplen que su valor es el doble del valor del elemento situado a su izquierda. **La inducción se ha de realizar sobre las columnas de la matriz**". Ejemplos:

$$M[1..3][1..4] = \begin{bmatrix} 1 & 2 & 4 & 8 \\ 3 & 6 & 12 & 24 \\ 5 & 10 & 20 & 40 \end{bmatrix}$$ Solución: VERDADERO

$$M[1..3][1..4] = \begin{bmatrix} 1 & 2 & 4 & 8 \\ 3 & 6 & 12 & 24 \\ 5 & 10 & 15 & 40 \end{bmatrix}$$ Solución: FALSO

Se pide responder a los siguientes apartados de manera clara, concisa y sin ambigüedad:

a) Especificación formal de la función (15%)

$Q \equiv \{n, m \geq 1\}$

Función Evaluacion (M[1..n][1..m]:matriz de enteros) retorna (b:booleano)

$$R \equiv \left\{ b = (\forall i)\big((\forall j)(2 * M[\,i\,][\,j\,] = M[\,i\,][\,j+1\,] : 1 \leq j \leq m-1) : 1 \leq i \leq n\big) \right\}$$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada matriz le asociaremos un natural m que corresponde al número de columnas de la matriz. Seguidamente aplicaremos inducción débil sobre dicho natural.

c) Realizar una inmersión no final, escribir la especificación formal de la función inmersora e indicar cómo se ha de efectuar la primera llamada a la función (10%)

Sustituimos m por k en *R*, de ese modo resulta:

$Q' \equiv \{\ 1 \leq n \text{ y } 1 \leq k \leq m\}$

Función iEvaluacion (M[1..n][1..m]:matriz de enteros, k:entero) retorna (b:booleano)

$$R' \equiv \left\{ b = (\forall i)\big((\forall j)(2 * M[\,i\,][\,j\,] = M[\,i\,][\,j+1\,] : 1 \leq j \leq k-1) : 1 \leq i \leq n\big) \right\}$$

La llamada inicial a la función iEvaluacion se realiza con k = m, esto es,

Evaluacion (M) = iEvaluacion (M, m)

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(M, k), para pasar al siguiente problema "eliminamos" una columna. Así, el subproblema sucesor será iEvaluacion(M, k-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

$B_t \equiv$ k = 1 la sección de la matriz tiene un única columna.

$B_{nt} \equiv$ k > 1 la sección de la matriz tiene varias columnas.

f) Explicar razonadamente la solución propuesta para el/los caso/s trivial/es, esto es, *triv*($\bar{x}$) (15%)

Al haber una única columna (k = 1), esto quiere decir que cada fila de la matriz tiene una única componente por lo que la solución del problema será VERDADERO ya que el único elemento de cada fila cumple la propiedad al no tener un elemento a su izquierda. Esto se puede comprobar en *R* al sustituir k por 1:

$$(\forall i)\big((\forall j)(2 * M[\,i\,][\,j\,] = M[\,i\,][\,j+1\,] : 1 \leq j \leq 1-1) : 1 \leq i \leq n\big)$$

$$(\forall i)\big((\forall j)(2 * M[\,i\,][\,j\,] = M[\,i\,][\,j+1\,] : 1 \leq j \leq 0) : 1 \leq i \leq n\big)$$

Al tener un rango vacío el resultado sería VERDADERO (elemento neutro de la operación que representa).

g) Explicar razonadamente la solución propuesta para el/los caso/s no trivial/es, esto es, explicar la función de combinación *c*($\bar{x}$, $\bar{y}'$) (25%)

La solución del problema iEvaluacion(M, k) consiste en determinar si se cumple la propiedad para la submatriz M[1..n][1..k], esto es,

$$(\forall i)\big((\forall j)(2 * M[\,i\,][\,j\,] = M[\,i\,][\,j+1\,] : 1 \leq j \leq k-1) : 1 \leq i \leq n\big)$$

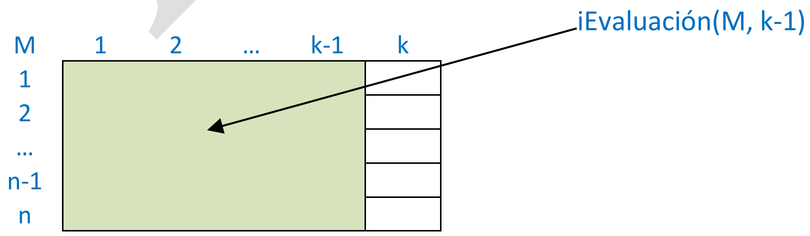
llamamos a dicho resultado, b.

Hipótesis de recurrencia: Supongamos conocida la solución del subproblema iEvaluacion(M, k-1). Llamaremos a dicho valor, b´. Esto quiere decir que suponemos conocida la solución para la submatriz M[1..n][1..k-1], esto es,

$$(\forall i)\big((\forall j)(2 * M[\,i\,][\,j\,] = M[\,i\,][\,j+1\,] : 1 \leq j \leq (k-1)-1) : 1 \leq i \leq n\big) \equiv$$

$$(\forall i)\big((\forall j)(2 * M[\,i\,][\,j\,] = M[\,i\,][\,j+1\,] : 1 \leq j \leq k-2) : 1 \leq i \leq n\big)$$

Gráficamente sería



Por lo que para obtener la solución del problema iEvaluacion(M, k) a partir de la solución del subproblema iEvaluacion(M,k-1) habría que comprobar además que cada elemento de la columna k sea el doble del elemento situado a su izquierda, esto es:

2*M[1][k-1] = M[1][k] y 2*M[2][k-1] = M[2][k], … y 2*M[n][k-1] = M[n][k]

Por tanto,

$$(\forall i)(2 * M[\,i\,][\,k-1\,] = M[\,i\,][\,k\,] : 1 \leq i \leq n) \qquad (1)$$

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (20%)

```
Función iEvaluacion ( M[1..n][1..m]: matriz de enteros; k : entero ) retorna
(b:booleano)
  caso
            k = 1 → VERDADERO
            k > 1 →  iEvaluacion ( M, k-1 ) AND AUX (M, k )
  fcaso
ffunción
```

La función AUX(M, k) es una función que calcula (1).-

```
Función AUX ( M[1..n][1..m]: matriz de enteros; k : entero ) retorna  ( b:booleano )
var i : entero, resultado : booleano fvar
i = 0
resultado = VERDADERO
mientras ( i < n AND resultado )
      i = i + 1
      si ( 2*M[ i ][ k-1 ] != M[ i ][ k ] ) entonces   resultado = FALSO  fsi
fmientras
retorna resultado
ffunción
```

2.- [3 puntos] Dada la siguiente función recursiva, se pide verificar formalmente su corrección:

Q' = { (1 ≤ i ≤ n) ^ (x > 0)}
Funcion iCalcula(V[1..n]: vector de enteros, n:entero, x:entero, i:entero) retorna (p: entero)
 si (i = 1) entonces retorna 1
 sino retorna iCalcula(V, n, x, i - 1) + V[i]*x^i
 fsi
ffuncion
R' = { p = (Σ k) (V[k] * x^k: 1 ≤ k ≤ i) }

La llamada inicial a la función será Calcula(V, n, x) = iCalcula(V, n, x, n).

Solución:

1) **Q($\overline{x}$) => Bt($\overline{x}$) ˅ Bnt($\overline{x}$)**

 (1 ≤ i ≤ n) ^ (x > 0) => (i = 1) ˅ (i > 1) Se cumple.

2) **Q($\overline{x}$) ^ Bnt($\overline{x}$) => Q(s($\overline{x}$))**

 (1 ≤ i ≤ n) ^ (x > 0) ^ (i > 1) => (i-1) ≥ 1 Se cumple.

3) **Q($\overline{x}$) ^ Bt($\overline{x}$) => R($\overline{x}$), triv($\overline{x}$))**
 (1 ≤ i ≤ n) ^ (x > 0) ^ (i = 1) => 1 = $(\sum k)$(V[k]*x^k : 1 ≤ k ≤ i)

 El antecedente de la implicación hace que el rango de k sea 1≤ k ≤ 1, lo cual conlleva a que k = 1, dando lugar a que el valor del cuantificador Σ sea igual a V[1]*x. Sin embargo, la función retorna 1 y por lo tanto no se cumpliría la postcondición (1 ≠ V[1]*x)

4) **Q($\overline{x}$) ^ Bnt($\overline{x}$) ^ R(s($\overline{x}$), $\overline{y}'$) => R($\overline{x}$, c(y', $\overline{x}$))**
 (1 ≤ i ≤ n) ^ (x > 0) ^ (i > 1) ^ (p' = $(\sum k)$(V[k]*x^k: 1 ≤ k ≤ i-1)) => p = p' + V[i]*x^i
 p = p' + V[i]*x^i = $(\sum k)$(V[k]*x^k: 1 ≤ k ≤ i-1) + V[i]*x^i = $(\sum k)$(V[k]*x^k : 1 ≤ k ≤ i)
 Por lo tanto, se cumple la postcondición.

5) **Encontrar la función limitadora t: D ->Z, tal que Q($\overline{x}$) => t($\overline{x}$) ≥ 0**

 t(V,n,x,i) $\stackrel{def}{=}$ i

 (1 ≤ i ≤ n) ^ (x > 0) => i ≥ 1 Se cumple.

6) **Q($\overline{x}$) ^ Bnt($\overline{x}$) => t(s($\overline{x}$)) < t($\overline{x}$)**

 (1 ≤ i ≤ n) ^ (x > 0) ^ (i > 1) => (i-1) < i Se cumple.

Curso Académico 2019-2020

MODELO A

1.- [8 puntos] Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:
"Dada una matriz A[1..n][1..n] de enteros, siendo n ≥ 1, diseñar una función recursiva que contabilice cuántas componentes de la diagonal, A[k][k] con k = 1, 2, …, n, cumplen que su valor es igual a

(Σ i)(A[i][k] * A[k][i] : 1 ≤ i ≤ k-1)"

$$A[1..3][1..3] = \begin{bmatrix} 0 & 4 & 2 \\ 5 & 20 & 2 \\ 1 & 3 & 8 \end{bmatrix}$$ Solución: 3

$$A[1..3][1..3] = \begin{bmatrix} 3 & 4 & 2 \\ 4 & 20 & 6 \\ 1 & 3 & 6 \end{bmatrix}$$ Solución: 0

Se pide responder a los siguientes apartados de manera clara, concisa y sin ambigüedad:

a) Especificación formal de la función (15%)

$Q \equiv \{ n \geq 1 \}$

Función Evaluacion (A[1..n][1..n]:matriz de enteros) retorna (b : entero)

$R \equiv \{ b = (\mathrm{N}k)(A[k][k] = (\Sigma\, i)(A[i][k] * A[k][i]: 1 \leq i \leq k-1): 1 \leq k \leq n)\}$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada matriz le asociaremos un natural que corresponde al orden de la matriz. Seguidamente aplicaremos inducción débil sobre dicho natural.

c) Realizar una inmersión no final y escribir la especificación formal de la función inmersora (10%)

Sustituimos n por j en *R*:

$Q' \equiv \{ \ 1 \leq j \leq n \}$

Función iEvaluacion (A[1..n][1..n]:matriz de enteros; j : entero) retorna (b:entero)

$R' \equiv \{ b = (\mathrm{N}k)(A[k][k] = (\Sigma\, i)(A[i][k] * A[k][i]: 1 \leq i \leq k-1): 1 \leq k \leq j)\}$

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(A, j), para pasar al siguiente eliminamos una fila y una columna. Así, el problema sucesor será iEvaluacion(A, j - 1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

B_t ≡ j = 1 — la sección de la matriz tiene un única fila y columna, esto es, un único elemento.

B_{nt} ≡ j > 1 — la sección de la matriz tiene varias filas y columnas, el mismo número de ambas.

f) Explicar razonadamente la solución propuesta para el caso trivial, esto es, *triv($\bar{x}$)* (15%)

En el caso de que la sección de la matriz tratada tuviera un único elemento (j = 1), la solución dependerá del valor de ese único elemento. Si dicho elemento, esto es, A[1][1], cumple que es igual a

(Σ i)(A[i][k] * A[k][i] : 1 ≤ i ≤ k-1)

la función retornará 1 y sino, 0. Veámoslo sobre R'

(N k)(A[k][k] = (Σ i)(A[i][k] * A[k][i] : 1 ≤ i ≤ k-1): 1 ≤ k ≤ **j**),

dado que estamos en el caso trivial, j es 1

(N k)(A[k][k] = (Σ i)(A[i][k] * A[k][i] : 1 ≤ i ≤ k-1): 1 ≤ k ≤ **1**), por tanto,

$$A[1][1] = (\Sigma\, i)(\, A[\,i\,][\,1\,] * A[\,1\,][\,i\,] : 1 \le i \le \mathbf{0}\,)$$

$$A[1][1] = 0$$

Entonces tendremos que el elemento A[1][1] debe ser igual a 0 ya que el sumatorio afecta a un rango vacío y su valor sería 0. Por lo que si A[1][1] es igual a 0, la solución sería 1. Si por el contrario el único elemento de la sección de matriz a tratar fuera distinto de 0, dicho componente de la diagonal no cumple que su valor es igual a la expresión indicada por lo que la función devolvería 0.

g) Explicar razonadamente la solución propuesta para el caso no trivial, esto es, explicar la función de combinación $c(\bar{x}, \bar{y'})$ (25%)

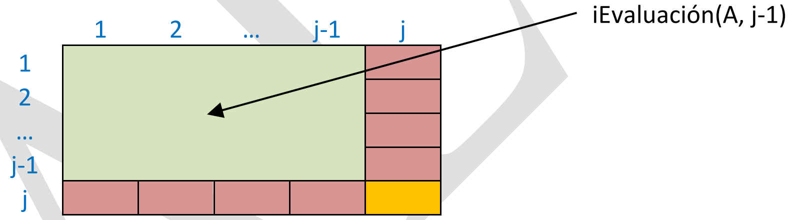
La solución del problema iEvaluacion(A, j) consiste en determinar cuántos elementos de la diagonal (intervalo 1..j) cumplen la condición pedida, esto es, cuántos A[k][k] con k=1, 2.., j, cumplen que son iguales a la expresión

$$(\Sigma\, i)(\, A[i][k] * A[k][i] : 1 \le i \le k-1\,)$$

Llamaremos a dicho resultado, b.

Dado que el problema sucesor es iEvaluacion(A, j-1). Esto quiere decir que suponemos conocido cuántos elementos de diagonal cumplen la condición pedida en el intervalo 1..j-1, esto es, cuántos A[k][k] con k=1, 2.., j-1, cumplen que son iguales a $(\Sigma\, i)(\, A[i][k] * A[k][i] : 1 \le i \le k-1\,)$. Llamaremos a dicho valor, b´ (en el dibujo que figura a continuación, correspondería con la zona verde).

Gráficamente sería



Para obtener la solución del problema iEvaluacion(A, j) a partir de la solución del subproblema iEvaluacion(A,j-1) bastaría comprobar si A[j][j] es igual a la siguiente expresión

$$(\Sigma\, i)(\, A[\,i\,][\,j\,] * A[\,j\,][\,i\,] : 1 \le i \le j-1\,).$$

Por tanto, si es igual sumaríamos 1 a la solución del problema sucesor, mientras que si fuera distinto sumaríamos 0 a la solución del problema sucesor,

$$b = b' + (\, A[\,j\,][\,j\,] = (\Sigma\, i)(\, A[\,i\,][\,j\,] * A[\,j\,][\,i\,] : 1 \le i \le j-1\,)\,)$$

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (10%)

```
Función iEvaluacion ( A[1..n][1..n]: matriz de enteros;  j : entero ) retorna  ( b:entero )
   si  j = 1 entonces
                  si A[ j ][ j ] = 0 entonces retorna 1
                       sino retorna 0
                  fsi
   sino retorna iEvaluacion ( A, j-1 ) + AUX ( A, j )
  fsi
función
```

La llamada inicial a la función iEvaluacion se realiza del siguiente modo: iEvaluacion(A, n)

donde la función AUX es.-

$Q = \{ 2 \leq j \leq n \}$

```
Función AUX ( A[1..n][1..n]: matriz de enteros;  j : entero) retorna  (b:entero)
var  i, s: entero fvar;
        s = 0
        para i = 1 hasta j-1 hacer
                 s = s + ( A[ i ][ j ] * A[ j ][ i ] )
        fpara
    si ( s = A[ j] [ j ] ) retorna 1   sino  retorna  0   fsi
ffuncion
```

$R \ \{ b = (A[j][j] = (\Sigma i)(A[i][j] * A[j][i] : 1 \leq i \leq j\text{-}1)) \}$

2.- [2 puntos] Responde a las siguientes cuestiones de manera clara, concisa y sin ambigüedad:

a) **[1 punto]** Demostrar los pasos 3 y 4 de verificación formal para la siguiente función:

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

```
Función iFuncion2 (V[1..n] : vector de enteros; x, k : entero ) retorna ( p : booleano)
          caso
                    k = n+1  → FALSO
                    k < n+1  → iFuncion2 (V, x, k + 1 ) OR (V[ k ] = x^k )
          fcaso
ffuncion
```

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

3. $Q(\overline{x}) \wedge B_t(\overline{x}) \Rightarrow R(\overline{x}, triv(\overline{x}))$

$1 \leq k \leq n+1 \wedge x \geq 0 \ \wedge k=n+1 \Rightarrow \ FALSO = (\exists i) (V[i] = x^i : k \leq i \leq n)$

Partimos de la postcondición, esto es,

$$p = (\exists i) (V[i] = x^i : k \leq i \leq n)$$

sustituimos p por FALSO y k por n+1 y tendremos

$$FALSO = (\exists i) (V[i] = x^i : n+1 \leq i \leq n) = FALSO$$

Sí se cumple, ya que cuando k=n+1 tendremos que el cuantificador ∃ afecta a un rango vacío, por lo que su valor es FALSO (elemento neutro de la operación que representa), y dicho valor es precisamente el que devuelve la función para el caso trivial.

4. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \wedge R(s(\overline{x}), \overline{y}') \Rightarrow R(\overline{x}, c(\overline{y}', \overline{x}))$

$1 \leq k \leq n+1 \wedge x \geq 0 \wedge k < n+1 \wedge p' = (\exists i)(V[i] = x^i : k+1 \leq i \leq n) \Rightarrow p = p' \text{ OR } (V[k] = x^k)$

$p = p' \text{ OR } (V[k] = x^k) =$

sustituimos p' por su valor

$= (\exists i)(V[i] = x^i : k+1 \leq i \leq n) \text{ OR } (V[k] = x^k) =$

$= [(V[k+1] = x^{k+1}) \text{ OR } (V[k+2] = x^{k+2}) \text{ OR} \ldots \text{OR } (V[n] = x^n)] \text{ OR } (V[k] = x^k) =$

$= (\exists i)(V[i] = x^i : k \leq i \leq n)$

Sí, se cumple.

b) **[0.50 puntos]** ¿Cómo definirías la funcion t (función limitadora) si tuvieras que demostrar los apartados 5 y 6 de verificación formal?

$t(V,x,k) \stackrel{def}{=} n-k+1$

c) **[0.50 puntos]** Si en lugar de reemplazar la constante 1 por k, como se hizo en el apartado a), hubiéramos reemplazado la constante n por k con objeto de que "se elimine" la última componente del vector para pasar al problema sucesor, ¿cuál sería/n el/los caso/s o trivial/es?

$k = 0$

MODELO B

1.- [8 puntos] Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:
"Dada una matriz A[1..n][1..n] de enteros, siendo n ≥ 1, diseñar una función recursiva que contabilice cuántas componentes de la diagonal, A[k][k] con k = 1, 2, …, n, cumplen que su valor es igual a
$(\Pi\, i)(A[i][k] + A[k][i] : 1 \leq i \leq k-1)$"

$A[1..3][1..3] = \begin{bmatrix} 1 & 4 & 2 \\ 5 & 9 & 2 \\ 1 & 3 & 15 \end{bmatrix}$ Solución: 3

$A[1..3][1..3] = \begin{bmatrix} 3 & 4 & 2 \\ 5 & 12 & 2 \\ 1 & 3 & 6 \end{bmatrix}$ Solución: 0

Se pide responder a los siguientes apartados de manera clara, concisa y sin ambigüedad:

a) Especificación formal de la función (15%)

$Q \equiv \{ n \geq 1 \}$

Función Evaluacion (A[1..n][1..n]:matriz de enteros) retorna (b : entero)

$R \equiv \{ b = (Nk)(A[k][k] = (\Pi\, i)(A[i][k] + A[k][i] : 1 \leq i \leq k-1) : 1 \leq k \leq n) \}$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada matriz le asociaremos un natural que corresponde al orden de la matriz. Seguidamente aplicaremos inducción débil sobre dicho natural.

c) Realizar una inmersión no final, escribir la especificación formal de la función inmersora e indicar cómo se ha de efectuar la primera llamada a la función (10%)

Sustituimos n por j en *R*:

$Q' \equiv \{\ 1 \leq j \leq n\ \}$

Función iEvaluacion (A[1..n][1..n]:matriz de enteros; j : entero) retorna (b:entero)

$R' \equiv \{\, b = (\mathrm{N}k)(A[k][k] = (\Pi\, i)(A[i][k] + A[k][i]: 1 \leq i \leq k-1): 1 \leq k \leq j)\}$

d) Identificar el sucesor o sucesores del problema dado (5%)

Dado el problema iEvaluacion(A, j), para pasar al siguiente eliminamos una fila y una columna. Así, el problema sucesor será iEvaluacion(A, j-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

Bt ≡ j = 1 la sección de la matriz tiene un única fila y columna, esto es, un único elemento.

Bnt ≡ j > 1 la sección de la matriz tiene varias filas y columnas, el mismo número de ambas.

f) Explicar razonadamente la solución propuesta para el/los caso/s trivial/es, esto es, *triv*($\bar{x}$) (15%)

En el caso de que la sección de la matriz tratada tuviera un único elemento (j = 1), la solución dependerá del valor de ese único elemento. Si dicho elemento, esto es, A[1][1], cumple que es igual a

(Π i)(A[i][k] + A[k][i] : 1 ≤ i ≤ k-1)

la función retornará 1 y sino, 0. Veámoslo sobre R'

(N k)(A[k][k] = (Π i)(A[i][k] + A[k][i] : 1 ≤ i ≤ k-1): 1 ≤ k ≤ **j**),

dado que estamos en el caso trivial, j es 1

(N k)(A[k][k] = (Π i)(A[i][k] + A[k][i] : 1 ≤ i ≤ k-1): 1 ≤ k ≤ **1**),

por tanto,

A[1][1] = (Π i)(A[i][1] + A[1][i] : 1 ≤ i ≤ **0**) = 1

A[1][1] = 1

Entonces tendremos que el elemento A[1][1] debe ser igual a 1 ya que Π afecta a un rango vacío y su valor sería 1. Por lo que si A[1][1] es igual a 1, la solución sería 1. Si por el contrario el único elemento de la sección de matriz a tratar fuera distinto de 1, dicho componente de la diagonal no cumple que su valor es igual a la expresión indicada, por lo que la función devolvería 0.

g) Explicar razonadamente la solución propuesta para el caso no trivial, esto es, explicar la función de combinación *c*($\bar{x}$, $\overline{y'}$) (25%)

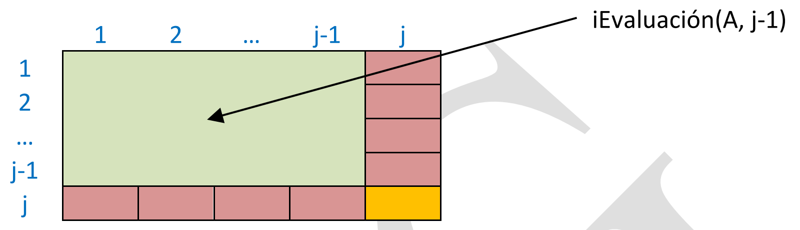
La solución del problema iEvaluacion(A, j) consiste en determinar cuántos A[k][k] con k=1, 2.., j, cumplen que son iguales a la expresión

$$(\Pi\ i)(\ A][i][k] + A[k][i] : 1 \le i \le k\text{-}1\ )$$

Llamaremos a dicho resultado, b.

Dado que el problema sucesor es iEvaluacion(A, j-1). Esto quiere decir que suponemos conocido cuántos elementos de diagonal cumplen la condición pedida en el intervalo 1..j-1, esto es, cuántos A[k][k] con k=1, 2.., j-1, cumplen que son iguales a $(\Pi\ i)(\ A][i][k] + A[k][i] : 1 \le i \le k\text{-}1\ )$. Llamaremos a dicho valor, b´ (en el dibujo que figura a continuación, correspondería con la zona verde).

Gráficamente sería



Para obtener la solución del problema iEvaluacion(A, j) a partir de la solución del subproblema iEvaluacion(A,j-1) bastaría comprobar si A[j][j] es igual a la siguiente expresión

$$(\Pi\ i)(\ A][\ i\ ][\ j\ ] + A[\ j\ ][\ i\ ] : 1 \le i \le j\text{-}1\ ).$$

Por tanto, si es igual sumaríamos 1 a la solución del problema sucesor, mientras que si fuera distinto sumaríamos 0 a la solución del problema sucesor,

$$b = b´ + (\ A[\ j\ ][\ j\ ] = (\Pi\ i)(\ A][\ i\ ][\ j\ ] + A[\ j\ ][\ i\ ] : 1 \le i \le j\text{-}1\ ))$$

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (20%)

```
Función iEvaluacion ( A[1..n][1..n]: matriz de enteros;  j : entero ) retorna  ( b:entero )
    si  j = 1 entonces
                      si A[ j ][ j ] = 1 entonces retorna 1
                               sino retorna 0
                      fsi
      sino retorna iEvaluacion ( A, j-1 ) + AUX ( A, j )
    fsi
ffunción
```

La llamada inicial a la función iEvaluacion se realiza del siguiente modo: iEvaluacion(A, n) donde la función AUX es.-

```
Q = { 2 ≤  j ≤ n }
Función AUX ( A[1..n][1..n]: matriz de enteros;  j : entero) retorna  (b:entero)
var  i, s: entero fvar;
        s = 1
        para i = 1 hasta j-1 hacer
                 s = s * ( A[ i ][ j ] + A[ j ][ i ] )
        fpara
    si ( s = A[ j] [ j ] ) retorna 1   sino  retorna  0   fsi
ffuncion
R  { b = ( A[ j ][ j ] = ( Π i )( A[ i ][ j ] + A[ j ][ i ] : 1 ≤ i ≤ j-1 )) }
```

2.- [2 puntos] Responde a las siguientes cuestiones de manera clara, concisa y sin ambigüedad:

a) **[1 punto]** Demostrar los pasos 3 y 4 de verificación formal para la siguiente función:

$Q' \equiv \{ 1 \leq k \leq n+1 \wedge x \geq 0 \}$
Función iFuncion2 (V[1..n] : vector de enteros; x, k : entero) retorna (p : booleano)
caso
$k = n+1 \rightarrow$ VERDADERO
$k < n+1 \rightarrow$ iFuncion2 (V, x, k + 1) AND (V[k] = x^k)
fcaso
ffuncion
$R' \equiv \{ p = (\forall i)(V[i] = x^i : k \leq i \leq n) \}$

3. $Q(\overline{x}) \wedge B_t(\overline{x}) \Rightarrow R(\overline{x}, triv(\overline{x}))$

$1 \leq k \leq n+1 \wedge x \geq 0 \;\wedge k=n+1 \Rightarrow$ VERDADERO = $(\forall i)(V[i] = x^i : k \leq i \leq n)$

Partimos de la postcondición, esto es,

$$p = (\forall i)(V[i] = x^i : k \leq i \leq n)$$

sustituimos p por VERDADERO y k por n+1 y tendremos

$$\text{VERDADERO} = (\forall i)(V[i] = x^i : n+1 \leq i \leq n) = \text{VERDADERO}$$

Sí se cumple, ya que cuando k=n+1 tendremos que el cuantificador ∀ afecta a un rango vacío, por lo que su valor es VERDADERO (elemento neutro de la operación que representa), y dicho valor es precisamente el que devuelve la función para el caso trivial.

4. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \wedge R(s(\overline{x}), \overline{y}') \Rightarrow R(\overline{x}, c(\overline{y}', \overline{x}))$

$1 \leq k \leq n+1 \wedge x \geq 0 \;\wedge k < n+1 \wedge p' = (\forall i)(V[i] = x^i : k+1 \leq i \leq n) \Rightarrow p = p'$ AND $(V[k] = x^k)$

$p = p'$ AND $(V[k] = x^k) =$

sustituimos p' por su valor

$= (\forall i)(V[i] = x^i : k+1 \leq i \leq n)$ AND $(V[k] = x^k) =$

$= [(V[k+1] = x^{k+1})$ AND $(V[k+2] = x^{k+2})$ AND ... AND $(V[n] = x^n)]$ AND $(V[k] = x^k) =$

$= (\forall i)(V[i] = x^i : k \leq i \leq n)$

Sí, se cumple.

b) **[0.50 puntos]** ¿Cómo definirías la funcion t (función limitadora) si tuvieras que demostrar los apartados 5 y 6 de verificación formal?

$$t(V,x,k) \stackrel{\text{def}}{=} n-k+1$$

c) **[0.50 puntos]** Si en lugar de reemplazar la constante 1 por k, como se hizo en el apartado a), hubiéramos reemplazado la constante n por k con objeto de que "se elimine" la última componente del vector para pasar al problema sucesor, ¿cuál sería/n el/los caso/s o trivial/es?

$$k = 0$$

Curso Académico 2018-2019

MODELO A

1.- [4 puntos] Se quiere diseñar una función recursiva que resuelva el siguiente problema:
"Dada una matriz A[1..n][1..m] siendo n y m mayores que 0, dicha función tiene que comprobar si los elementos de cada fila, leídos de izquierda a derecha, están en orden estrictamente decreciente. **La inducción se ha de realizar sobre las columnas de la matriz**". Ejemplos:

$$A[1..3][1..4] = \begin{bmatrix} 7 & 5 & 4 & 3 \\ 12 & 8 & 5 & 2 \\ 81 & 9 & 2 & 1 \end{bmatrix}$$ Solución: VERDADERO

$$A[1..3][1..4] = \begin{bmatrix} 7 & 1 & 4 & 3 \\ 12 & 8 & 5 & 2 \\ 81 & 9 & 2 & 1 \end{bmatrix}$$ Solución: FALSO

Se pide responder a los siguientes apartados de manera clara, concisa y sin ambigüedad:

SOLUCIÓN:

a) Especificación formal de la función (15%)

$Q \equiv \{n, m \geq 1\}$

Función Evaluacion (A[1..n][1..m]:matriz de enteros) retorna (b:booleano)

$R \equiv \left\{b = (\forall i)\big((\forall j)(A[\,i\,][\,j\,] > A[\,i\,][\,j+1\,] : 1 \leq j \leq m-1) : 1 \leq i \leq n\big)\right\}$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada matriz le asociaremos un natural m que corresponde al número de columnas de la matriz. Seguidamente aplicaremos inducción débil sobre dicho natural.

c) Realizar una inmersión no final, escribir la especificación formal de la función inmersora e indicar cómo se ha de efectuar la primera llamada a la función (10%)

Sustituimos m por k en *R*:

$Q' \equiv \{\ 1 \leq n\ y\ 1 \leq k \leq m\}$

Función iEvaluacion (A[1..n][1..m]:matriz de enteros, k:entero) retorna (b:booleano)

$R' \equiv \left\{b = (\forall i)\big((\forall j)(A[\,i\,][\,j\,] > A[\,i\,][\,j+1\,] : 1 \leq j \leq k-1) : 1 \leq i \leq n\big)\right\}$

La llamada inicial a la función iEvaluacion se realiza con k = m, esto es,

Evaluacion (A) = iEvaluacion (A, m)

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(A, k), para pasar al siguiente problema eliminamos una columna. Así, el subproblema sucesor será iEvaluacion(A, k-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

$B_t \equiv k = 1$ la sección de la matriz tiene un única columna.

$B_{nt} \equiv k > 1$ la sección de la matriz tiene varias columnas.

f) Explicar razonadamente la solución propuesta para el/los caso/s trivial/es, esto es, *triv($\bar{x}$)* (15%)

En el caso de que la sección de la matriz tratada tenga una única columna (k = 1), esto quiere decir que cada fila de la matriz tiene un único componente por lo que la solución del problema será VERDADERO ya que el único elemento de cada fila cumple la propiedad.

g) Explicar razonadamente la solución propuesta para el/los caso/s no trivial/es, esto es, explicar la función de combinación *c($\bar{x}$, $\bar{y}'$)* (25%)

La solución del problema iEvaluacion(A, k) consiste en determinar si se cumple la propiedad para la submatriz A[1..n][1..k], esto es,

$$(\forall i)\big((\forall j)(A[\,i\,][\,j\,] > A[\,i\,][\,j+1\,] : 1 \le j \le k-1) : 1 \le i \le n\big)$$

llamamos a dicho resultado, b.

Hipótesis de recurrencia: Supongamos conocida la solución del subproblema iEvaluacion(A, k-1). Llamaremos a dicho valor, b´. Esto quiere decir que suponemos conocida la solución para la submatriz A[1..n][1..k-1], esto es,

$$(\forall i)\big((\forall j)(A[\,i\,][\,j\,] > A[\,i\,][\,j+1\,] : 1 \le j \le (k-1)-1) : 1 \le i \le n\big) \equiv$$

$$(\forall i)\big((\forall j)(A[\,i\,][\,j\,] > A[\,i\,][\,j+1\,] : 1 \le j \le k-2) : 1 \le i \le n\big)$$

Por lo que para obtener la solución del problema iEvaluacion(A, k) a partir de la solución del subproblema iEvaluacion(A,k-1) faltaría comprobar para cada fila que el elemento de la columna k es menor estricto que el de la columna k-1, esto es

A[1][k-1] > A[1][k], A[2][k-1] > A[2][k], …., A[n][k-1] > A[n][k], es decir,

$$(\forall i)(A[\,i\,][\,k-1\,] > A[\,i\,][\,k\,] : 1 \le i \le n) \qquad (1)$$

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (20%)

```
Función iEvaluacion ( A[1..n][1..m]: matriz de enteros; k : entero ) retorna (b:booleano)
  caso
        k = 1 → VERDADERO
        k > 1 →  iEvaluacion ( A, k-1 ) AND AUX (A, k )
  fcaso
ffunción
```

donde AUX(A, k) es una función que calcula (1).

```
Función AUX ( A[1..n][1..m]: matriz de enteros; k : entero ) retorna  ( b:booleano )
var i : entero, resultado : booleano fvar
i = 0
resultado = VERDADERO
mientras ( i < n AND resultado )
    i = i + 1
    si ( A[ i ][ k-1 ] <= A[ i ][ k ] ) entonces   resultado = FALSO  fsi
fmientras
retorna resultado
ffunción
```

2.- [1 punto] Responde a las siguientes cuestiones de manera clara, concisa y sin ambigüedad:

a) [0.25 puntos] Si aplicamos recursión sobre un vector V[1..n] con $n \geq 1$, reemplazando en la postcondición la constante 1 por k con objeto de que "se elimine" la primera componente del vector para pasar al problema sucesor, ¿cuál es la función t (función limitadora) que definirías para demostrar los apartados 5 y 6 de verificación formal?

$$t(V,k) \stackrel{\text{def}}{=} n-k+1$$

b) [0.25 puntos] Si aplicamos recursión sobre un vector V[1..n] con $n \geq 0$, reemplazando en la postcondición la constante 1 por la variable p y la constante n por la variable q con objeto de que "se eliminen" la primera y última componente del vector para pasar al problema sucesor, ¿cuál/es será/n el/los caso/s o trivial/es?

$$p > q$$

c) [0.50 puntos] Demostrar los pasos 3 y 4 de verificación formal para la siguiente función:

```
{ 1 ≤ j ≤ n}
Function iF2c (V[1..n]: vector de enteros; j:entero) retorna (s: entero)
 si j == 1 entonces retorna 0;
   sino retorna iF2c ( V, j-1 ) + (V[ j - 1] == V[ j ]);
 fsi
ffuncion
{ s = ( N i )( V[ i ] == V[ i + 1 ] : 1 ≤ i < j ) }
```

3. $Q(\overline{x}) \wedge B_t(\overline{x}) \Rightarrow R(\overline{x}, triv(\overline{x}))$

$1 \leq j \leq n \wedge j = 1 \Rightarrow 0 = (\text{N } i)\,(V[i] == V[i+1] : 1 \leq i < j)$

Si j=1 entonces $(\text{N } i)\,(V[i] == V[i+1] : 1 \leq i < j) = (\text{N } i)\,(V[i] == V[i+1] : 1 \leq i < 1)$ lo que quiere decir que el cuantificador N afecta a un rango vacío, por lo que su valor es 0 (elemento neutro de la operación que representa), siendo efectivamente ese valor (0) el que devuelve la función para el caso trivial.

4. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \wedge R(s(\overline{x}), \overline{y}') \Rightarrow R(\overline{x}, c(\overline{y}', \overline{x}))$

$1 \leq j \leq n \wedge j > 1 \wedge s' = (\text{N } i)\,(V[i] == V[i+1] : 1 \leq i < j-1) \Rightarrow s = s' + (V[j-1] == V[j])$

$s = s' + (V[j-1] == V[j]) = (\text{N } i)\,(V[i] == V[i+1] : 1 \leq i < j-1) + (V[j-1] == V[j]) =$

$= [\,(V[1] == V[2]) + (V[2] == V[3]) + (V[3] == V[4]) + \ldots + (V[j-2] == V[j-1])\,] + (V[j-1] == V[j]) =$

$= (\text{N } i)\,(V[i] == V[i+1] : 1 \leq i < j)$

Sí se cumple.

MODELO B

1.- [4 puntos] Se quiere diseñar una función recursiva que resuelva el siguiente problema:
"Dada una matriz B[1..n][1..m] siendo n y m mayores que 0, dicha función tiene que comprobar si los elementos de cada columna, leídos de arriba abajo, están en orden estrictamente creciente. **La inducción se ha de realizar sobre las filas de la matriz**". Ejemplos:

$$B[1..4][1..3] = \begin{bmatrix} 6 & 1 & 7 \\ 9 & 0 & 8 \\ 15 & 5 & 34 \\ 45 & 7 & 56 \end{bmatrix}$$ Solución: FALSO

$$B[1..4][1..3] = \begin{bmatrix} 6 & 1 & 7 \\ 9 & 4 & 8 \\ 15 & 5 & 34 \\ 45 & 7 & 56 \end{bmatrix}$$ Solución: VERDADERO

Se pide responder a los siguientes apartados de manera clara, concisa y sin ambigüedad:

SOLUCIÓN:

a) Especificación formal de la función (15%)

$Q \equiv \{n, m \geq 1\}$

Función Evaluacion (B[1..n][1..m]:matriz de enteros) retorna (b:booleano)

$$R \equiv \{b = (\forall j)((\forall i)(B[\,i\,][\,j\,] < B[\,i+1\,][\,j\,] : 1 \leq i \leq n-1) : 1 \leq j \leq m)\}$$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada matriz le asociaremos un natural n que corresponde al número de filas de la matriz. Seguidamente aplicaremos inducción débil sobre dicho natural.

c) Realizar una inmersión no final, escribir la especificación formal de la función inmersora e indicar cómo se ha de efectuar la primera llamada a la función (10%)

Sustituimos n por k en *R*:

$Q' \equiv \{\ 1 \leq m\ y\ 1 \leq k \leq n\}$

Función iEvaluacion (B[1..n][1..m]:matriz de enteros, k:entero) retorna (b:booleano)

$$R' \equiv \{b = (\forall j)((\forall i)(B[\,i\,][\,j\,] < B[\,i+1\,][\,j\,] : 1 \leq i \leq k-1) : 1 \leq j \leq m)\}$$

La llamada inicial a la función iEvaluacion se realiza con k = n, esto es,

Evaluacion (B) = iEvaluacion (B, n)

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(B, k), para pasar al siguiente problema eliminamos una fila. Así, el subproblema sucesor será iEvaluacion(B, k-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

$B_t \equiv$ k = 1 la sección de la matriz tiene un única fila.

$B_{nt} \equiv$ k > 1 la sección de la matriz tiene varias filas.

f) Explicar razonadamente la solución propuesta para el/los caso/s trivial/es, esto es, *triv*($\bar{x}$) (15%)

En el caso de que la sección de la matriz tratada tenga una única fila (k = 1), esto quiere decir que cada columna de la matriz tiene un único componente por lo que la solución del problema será VERDADERO ya que el único elemento de cada columna cumple la propiedad.

g) Explicar razonadamente la solución propuesta para el/los caso/s no trivial/es, esto es, explicar la función de combinación $c(\bar{x}, \overline{y'})$ (25%)

La solución del problema iEvaluacion(B, k) consiste en determinar si se cumple la propiedad para la submatriz B[1..k][1..m], esto es,

$$(\forall j)\big((\forall i)(B[\,i\,][\,j\,] < B[\,i+1\,][\,j\,] : 1 \le i \le k-1) : 1 \le j \le m\big)$$

llamamos a dicho resultado, b.

Hipótesis de recurrencia: Supongamos conocida la solución del subproblema iEvaluacion(B, k-1). Llamaremos a dicho valor, b´. Esto quiere decir que suponemos conocida la solución para la submatriz B[1..k-1][1..m], esto es,

$$(\forall j)\big((\forall i)(B[\,i\,][\,j\,] < B[\,i+1\,][\,j\,] : 1 \le i \le (k-1)-1) : 1 \le j \le m\big) \equiv$$

$$(\forall j)\big((\forall i)(B[\,i\,][\,j\,] < B[\,i+1\,][\,j\,] : 1 \le i \le k-2) : 1 \le j \le m\big)$$

Por lo que para obtener la solución del problema iEvaluacion(B, k) a partir de la solución del subproblema iEvaluacion(B,k-1) faltaría comprobar para cada columna que el elemento de la fila k es mayor estricto que el de la fila k-1, esto es

B[k-1][1] < B[k][1], B[k-1][2] < B[k][2], …., B[k-1][m] < B[k][m], es decir,

$$(\forall j)(B[\,k-1\,][\,j\,] < B[\,k\,][\,j\,] : 1 \le j \le m) \qquad (1)$$

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (20%)

```
Función iEvaluacion ( B[1..n][1..m]: matriz de enteros; k : entero ) retorna (b:booleano)
  caso
           k = 1 → VERDADERO
           k > 1 → iEvaluacion ( B, k-1 ) AND AUX (B, k )
  fcaso
ffunción
```

donde AUX(B, k) es una función que calcula (1).

```
Función AUX ( B[1..n][1..m]: matriz de enteros; k : entero ) retorna ( b:booleano )
var i : entero, resultado : booleano fvar
j = 0
resultado = VERDADERO
mientras ( j < m AND resultado )
    j = j + 1
    si ( B[ k-1 ][ j ] >= B[ k ][ j ] ) entonces  resultado = FALSO  fsi
fmientras
retorna resultado
ffunción
```

2.- [1 punto] Responde a las siguientes cuestiones de manera clara, concisa y sin ambigüedad:

a) [0.25 puntos] Si aplicamos recursión sobre un vector V[1..n] con $n \geq 1$, reemplazando en la postcondición la constante 1 por q con objeto de que "se elimine" la primera componente del vector para pasar al problema sucesor, ¿cuál es la función t (función limitadora) que definirías para demostrar los apartados 5 y 6 de verificación formal?

$$t(V,k) \stackrel{def}{=} n-k+1$$

b) [0.25 puntos] Si aplicamos recursión sobre un vector V[1..n] con $n \geq 1$, reemplazando en la postcondición la constante 1 por la variable p y la constante n por la variable q con objeto de que "se eliminen" la primera y última componente del vector para pasar al problema sucesor, ¿cuál/es será/n el/los caso/s o trivial/es?

p == q y p == q-1

c) [0.50 puntos] Demostrar los pasos 3 y 4 de verificación formal para la siguiente función:

```
{ 1 ≤ k ≤ n}
Function iF2c (V[1..n]: vector de enteros; k:entero) retorna (s: entero)
 si k == n entonces retorna 0;
   sino retorna iF2c ( V, k+1 ) + (V[ k ] == V[ k + 1 ]);
 fsi
 ffuncion
{ s = ( N i )( V[ i ] == V[ i + 1 ] : k ≤ i < n ) }
```

3. $Q(\bar{x}) \wedge B_t(\bar{x}) \Rightarrow R(\bar{x}, triv(\bar{x}))$

$1 \leq k \leq n \wedge k = n \Rightarrow 0 = (\,N\, i\,)\,(\,V[\,i\,] == V[\,i + 1]: k \leq i < n\,)$

Si k=n entonces $(\,N\, i\,)\,(\,V[\,i\,] == V[\,i + 1]: k \leq i < n\,) = (\,N\, i\,)\,(\,V[\,i\,] == V[\,i + 1]: n \leq i < n\,)$ lo que quiere decir que el cuantificador N afecta a un rango vacío, por lo que su valor es 0 (elemento neutro de la operación que representa), siendo efectivamente ese valor (0) el que devuelve la función para el caso trivial.

4. $Q(\bar{x}) \wedge B_{nt}(\bar{x}) \wedge R(s(\bar{x}), \bar{y}') \Rightarrow R(\bar{x}, c(\bar{y}', \bar{x}))$

$1 \leq k \leq n \wedge k < n \wedge s' = (\,N\, i\,)\,(\,V[\,i\,] == V[\,i + 1]: k+1 \leq i < n) \Rightarrow s = s' + (V[\,k\,] == V[\,k+1\,])$

$s = s' + (V[\,k\,] == V[\,k+1\,]) = (\,N\, i\,)\,(\,V[\,i\,] == V[\,i + 1]: k+1 \leq i < n) + (V[\,k\,] == V[\,k+1\,]) =$

$= [(V[k+1] == V[k+2]) + (V[k+2] == V[k+3]) + \ldots + (V[n-1] == V[n])\,] + (V[\,k\,] == V[\,k+1\,]) =$

$= (\,N\, i\,)\,(\,V[\,i\,] == V[\,i + 1]: k \leq i < n\,)$

Sí se cumple.

Curso Académico 2017-2018

MODELO A

1.- [8 puntos] Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:
"Dada una matriz A[1..n][1..n] de enteros, siendo n ≥ 1, diseñar una función recursiva que determine si todos los elementos por encima de la diagonal principal son pares y todos los elementos por debajo de la diagonal principal son impares." Ejemplos:

$$A[1..3][1..3] = \begin{bmatrix} 0 & 4 & 2 \\ 5 & 5 & 6 \\ 1 & 3 & 8 \end{bmatrix}$$ retornará VERDADERO

$$A[1..3][1..3] = \begin{bmatrix} 0 & 4 & 2 \\ 4 & 5 & 6 \\ 1 & 3 & 8 \end{bmatrix}$$ retornará FALSO

Se pide responder a los siguientes apartados de manera clara, concisa y sin ambigüedad:

a) Especificación formal de la función (15%)

$Q \equiv \{\, n \geq 1 \,\}$

Función Evaluacion (A[1..n][1..n]:matriz de enteros) retorna (b:booleano)

$$R \equiv \left\{ b = (\forall i)\big((\forall j)(A[\,i\,][\,j\,]\%2 \neq 0 \;\; y \;\; A[\,j\,][\,i\,]\%2 = 0 : 1 \leq j \leq i-1) : 1 \leq i \leq n\big) \right\}$$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada matriz le asociaremos un natural n que corresponde al orden de la matriz. Seguidamente aplicaremos inducción débil sobre dicho natural n.

c) Realizar una inmersión no final, escribir la especificación formal de la función inmersora e indicar cómo se ha de efectuar la primera llamada a la función (10%)

Sustituimos n por k en R:

$Q' \equiv \{\; 1 \leq k \leq n \,\}$

Función iEvaluacion (A[1..n][1..n]:matriz de enteros, k:entero) retorna (b:booleano)

$$R' \equiv \left\{ b = (\forall i)\big((\forall j)(A[\,i\,][\,j\,]\%2 \neq 0 \;\; y \;\; A[\,j\,][\,i\,]\%2 = 0 : 1 \leq j \leq i-1) : 1 \leq i \leq k\big) \right\}$$

La llamada inicial a la función iEvaluacion se realiza con k = n, esto es,

Evaluacion (A) = iEvaluacion (A, n)

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(A, k), para pasar al siguiente problema eliminamos una fila y una columna. Así, el subproblema sucesor será iEvaluacion(A, k-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

$B_t \equiv$ k = 1 la sección de la matriz tiene un única fila y columna, esto es, un único elemento.

$B_{nt} \equiv$ k > 1 la sección de la matriz tiene varias filas y columnas, el mismo número de ambas.

f) Explicar razonadamente la solución propuesta para el/los caso/s trivial/es, esto es, *triv*($\bar{x}$) (15%)

En el caso de que la sección de la matriz tratada tuviera un único elemento (k = 1), ese único elemento pertenece a la diagonal principal, por lo que al no existir más elementos (ni por encima ni por debajo de la diagonal) la solución del problema será VERDADERO ya que las secciones a tratar serán vacías.

g) Explicar razonadamente la solución propuesta para el/los caso/s no trivial/es, esto es, explicar la función de combinación $c(\bar{x}, \bar{y}')$ (25%)

La solución del problema iEvaluacion(A, k) consiste en determinar si todos los elementos por encima de la diagonal principal de la sección de la matriz A[1..k][1..k] son pares y si todos los elementos por debajo de la diagonal principal de la sección de la matriz A[1..k][1..k] son impares, esto es,

$$(\forall i)((\forall j)\ (A[\,i\,][\,j\,]\ \%\ 2 \neq 0 \text{ y } A[\,j\,][\,i\,]\ \%\ 2 = 0: 1 \leq j \leq i\text{-}1\ ): 1 \leq i \leq k),$$

llamamos a dicho resultado, b.

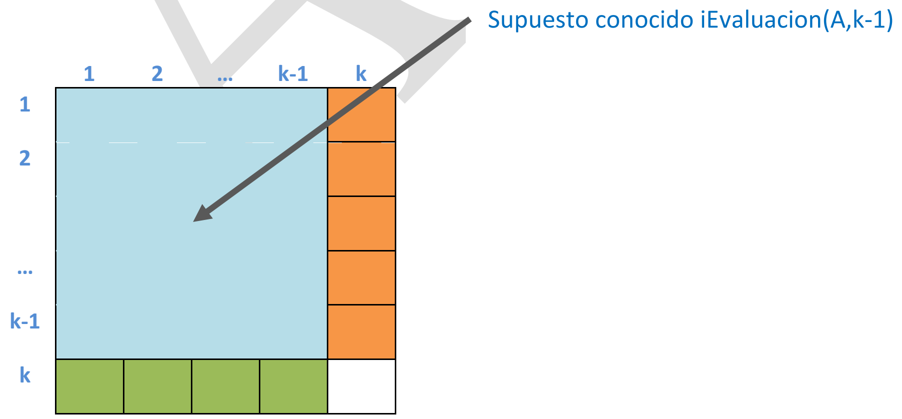
Hipótesis de recurrencia: Supongamos conocida la solución del subproblema iEvaluacion(A, k-1). Llamaremos a dicho valor, b´. Esto quiere decir que suponemos conocido si todos los elementos por encima de la diagonal principal de la sección de la matriz A[1..k-1][1..k-1] son pares y si todos los elementos por debajo de la diagonal principal de la sección de la matriz A[1..k-1][1..k-1] son impares, esto es,

$$(\forall i)((\forall j)\ (A[\,i\,][\,j\,]\ \%\ 2 \neq 0 \text{ y } A[\,j\,][\,i\,]\ \%\ 2 = 0: 1 \leq j \leq i\text{-}1\ ): 1 \leq i \leq k\text{-}1),$$

Por lo que para obtener la solución del problema iEvaluacion(A, k) a partir de la solución del subproblema iEvaluacion(A,k-1) bastaría comprobar que los elementos A[k][1], A[k][2], …., A[k][k-1] son impares y que los elementos A[1][k], A[2][k], …., A[k-1][k] son pares, esto es,

$$(\forall j)\ (A[\,k\,][\,j\,]\ \%\ 2 \neq 0 \text{ y } A[\,j\,][\,k\,]\ \%\ 2 = 0: 1 \leq j \leq k\text{-}1\ ) \qquad (1)$$

Graficamente,



Por tanto, para obtener iEvaluacion(A,k) deberemos comprobar además si:

a) A[k][1] %2 ≠ 0 AND A[k][2] %2 ≠ 0 AND … AND A[k][k – 1] %2 ≠ 0 (en verde) y

b) A[1][k] %2 = 0 AND A[2][k] %2 = 0 AND … AND A[k – 1][k] %2 = 0 (en naranja)

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (20%)

```
Función iEvaluacion ( A[1..n][1..n]: matriz de enteros; k : entero ) retorna  ( b:booleano )
  caso
        k = 1 → VERDADERO
        k > 1 →  iEvaluacion ( A, k-1 ) AND AUX (A, k )
  fcaso
ffunción
```

La llamada inicial a la función iEvaluacion se realiza del siguiente modo: iEvaluacion(A, n).

La función AUX(A, k) es una función que calcula (1).-

```
Función AUX ( A[1..n][1..n]: matriz de enteros; k : entero ) retorna  ( b:booleano )
var i : entero, resul : booleano fvar
i = 0
resul = VERDADERO
mientras ( i < k-1 AND resul )
    i = i + 1
    si ( A[ i ][ k ] %2 ≠ 0 OR A[ k ][ i ] %2 = 0 ) entonces
        resul = FALSO
    fsi
fmientras
retorna resul
ffunción
```

2.- [2 puntos] Dada la siguiente función recursiva, se pide verificar formalmente su corrección:

$Q' \equiv \{ 0 \le k \le n \wedge x \ge 0 \}$

```
Función iF2 ( A[1..n] : vector de enteros, x : entero, k : entero  ) retorna ( p : entero )
    caso
        k = 0   → 0
        k > 0   → iF2 (A, x, k-1 )  +  ( A[ k ] + x^k )
    fcaso
ffuncion
```

$R' \equiv \{ p = (\Pi\, i)\,(A[i] + x^i : 1 \le i \le k) \}$

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

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

SOLUCIÓN:

1. $Q(\overline{x}) \Rightarrow B_t(\overline{x}) \vee B_{nt}(\overline{x})$

$0 \le k \le n \wedge x \ge 0 \Rightarrow k = 0 \vee k > 0$ Se cumple.

2. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \Rightarrow Q(s(\overline{x}))$

$0 \le k \le n \wedge x \ge 0 \wedge k > 0 \Rightarrow k-1 \ge 0$ Se cumple.

3. $Q(\overline{x}) \wedge B_t(\overline{x}) \Rightarrow R(\overline{x}, triv(\overline{x}))$
$0 \le k \le n \wedge x \ge 0 \wedge k = 0 \Rightarrow 0 = (\Pi\, i)(A[i] + x^i : 1 \le i \le k)$
No se cumple, ya que cuando k=0 tendremos que el cuantificador Π afecta a un rango vacío, por lo que su valor es 1 (elemento neutro de la operación producto). Sin embargo, la función iF2 devuelve 0. Para que se cumpla la poscondición en el caso trivial, la función debería retornar 1.

4. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \wedge R(s(\overline{x}), \overline{y}') \Rightarrow R(\overline{x}, c(\overline{y}', \overline{x}))$
$0 \le k \le n \wedge x \ge 0 \wedge k > 0 \wedge p' = (\Pi\, i)(A[i] + x^i : 1 \le i \le k\text{-}1) \Rightarrow p = p' + (A[k] + x^k)$

$p = p' + (A[k] + x^k) = (\Pi\, i)(A[i] + x^i : 1 \le i \le k\text{-}1) + (A[k] + x^k) \neq (\Pi\, i)(A[i] + x^i : 1 \le i \le k)$

No se cumple, ya que la función de combinación de iF2 consiste en sumar el término k-ésimo cuando en realidad debería multiplicar el término k-ésimo, esto es, $p = p' * (A[k] + x^k)$ ya que el cuantificador Π representa la operación producto $((A[1] + x^1) * (A[2] + x^2) * \ldots * (A[k] + x^k))$

5. Encontrar una función t:D → Z, tal que $Q(\overline{x}) \Rightarrow t(\overline{x}) \ge 0$
$t(A,x,k) \stackrel{\text{def}}{=} k$

$0 \le k \le n \wedge x \ge 0 \Rightarrow k \ge 0$ Se cumple.

6. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \Rightarrow t(s(\overline{x})) < t(\overline{x})$
$0 \le k \le n \wedge x \ge 0 \wedge k > 0 \Rightarrow k\text{-}1 < k$ Se cumple.

MODELO B

1.- [8 puntos] Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:
"Dada una matriz A[1..n][1..n] de enteros, siendo n ≥ 1, diseñar una función recursiva que determine si todos los elementos por encima de la diagonal principal cumplen que su valor coincide con la suma de los índices de la fila y de la columna en la que están situados y todos los elementos por debajo de la diagonal principal cumplen que su valor coincide con el producto de los índices de la fila y de la columna en la que están situados.". Ejemplos:

$A[1..3][1..3] = \begin{bmatrix} 0 & 3 & 4 \\ 2 & 5 & 5 \\ 3 & 6 & 8 \end{bmatrix}$ retornará VERDADERO

$A[1..3][1..3] = \begin{bmatrix} 0 & 3 & 4 \\ 2 & 5 & 5 \\ 5 & 6 & 8 \end{bmatrix}$ retornará FALSO

Se pide responder a los siguientes apartados de manera clara, concisa y sin ambigüedad:

a) Especificación formal de la función (15%)

$Q \equiv \{ n \ge 1 \}$

Función Evaluacion (A[1..n][1..n]:matriz de enteros) retorna (b:booleano)

$R \equiv \left\{ b = (\forall i)\big((\forall j)(A[i][j] = i * j \ y \ A[j][i] = i + j : 1 \le j \le i - 1) : 1 \le i \le n\big) \right\}$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada matriz le asociaremos un natural n que corresponde al orden de la matriz. Seguidamente aplicaremos inducción débil sobre dicho natural n.

c) Realizar una inmersión no final , escribir la especificación formal de la función inmersora e indicar cómo se ha de efectuar la primera llamada a la función (10%)

Sustituimos n por k en *R*:

$Q' \equiv \{\ 1 \leq k \leq n\ \}$

Función iEvaluacion (A[1..n][1..n]:matriz de enteros, k:entero) retorna (b:booleano)

$R' \equiv \left\{b = (\forall i)\big((\forall j)(A[i][j] = i * j\ \ y\ \ A[j][i] = i + j : 1 \leq j \leq i - 1) : 1 \leq i \leq k\big)\right\}$

La llamada inicial a la función iEvaluacion se realiza con k = n, esto es,

Evaluacion (A) = iEvaluacion (A, n)

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(A, k), para pasar al siguiente problema eliminamos una fila y una columna. Así, el subproblema sucesor será iEvaluacion(A, k-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

$B_t \equiv$ k = 1 la sección de la matriz tiene un única fila y columna, esto es, un único elemento.

$B_{nt} \equiv$ k > 1 la sección de la matriz tiene varias filas y columnas, el mismo número de ambas.

f) Explicar razonadamente la solución propuesta para el caso trivial, esto es, *triv*($\bar{x}$) (15%)

En el caso de que la sección de la matriz tratada tuviera un único elemento (k = 1), ese único elemento pertenece a la diagonal principal, por lo que al no existir más elementos (ni por encima ni por debajo de la diagonal) la solución del problema será VERDADERO ya que las secciones a tratar serán vacías.

g) Explicar razonadamente la solución propuesta para el caso no trivial, esto es, explicar la función de combinación *c*($\bar{x}$, $\bar{y}'$) (25%)

La solución del problema iEvaluacion(A,k) consiste en determinar si todos los elementos por encima de la diagonal principal de la sección de la matriz A[1..k][1..k] cumplen que su valor coincide con la suma de los índices de la fila y de la columna en la que están situados y si todos los elementos por debajo de la diagonal principal de la sección de la matriz A[1..k][1..k] cumplen que su valor coincide con el producto de los índices de la fila y de la columna en la que están situados, esto es,

$(\forall i)((\forall j)\ (A[\,i\,][\,j\,] = i * j\ \ y\ \ A[\,j\,][\,i\,] = i + j : 1 \leq j \leq i\text{-}1) : 1 \leq i \leq k),$

llamamos a dicho resultado, b.

Hipótesis de recurrencia: Supongamos conocida la solución del subproblema iEvaluacion(A,k-1). Llamaremos a dicho valor, b´. Esto quiere decir que suponemos conocido si todos los elementos por encima de la diagonal principal de la sección de la matriz A[1..k-1][1..k-1] cumplen que su valor coincide con la suma de los índices de la fila y de la columna en la que están situados y si todos los elementos por debajo de la diagonal principal de la sección de la matriz A[1..k-1][1..k-1] cumplen que su valor coincide con el producto de los índices de la fila y de la columna en la que están situados, esto es,

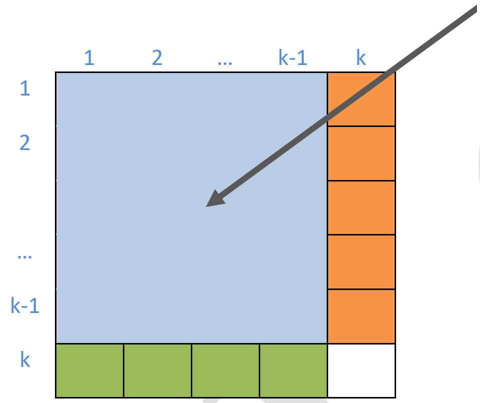
$$(\forall i)((\forall j)\ (A[\ i\ ][\ j\ ] = i * j \text{ y } A[\ j\ ][\ i\ ] = i + j : 1 \leq j \leq i\text{-}1): 1 \leq i \leq k\text{-}1),$$

Por lo que para obtener la solución del problema iEvaluacion(A, k) a partir de la solución del subproblema iEvaluacion(A,k-1) bastaría comprobar que los elementos A[k][1], A[k][2], …., A[k][k-1] cumplen que su valor coincide con el producto de los índices de la fila y de la columna en la que están situados y que los elementos A[1][k], A[2][k], …., A[k-1][k] cumplen que su valor coincide con la suma de los índices de la fila y de la columna en la que están situados, esto es,

$$(\forall j)\ (A[\ k\ ][\ j\ ] = i * j \text{ y } A[\ j\ ][\ k\ ] = i + j : 1 \leq j \leq k\text{-}1\ ) \qquad (1)$$

Graficamente,

Supuesto conocido iEvaluacion(A, k-1)



Por tanto, para obtener iEvaluacion(A,k) deberemos comprobar además si:

a) A[k][1] = k*1 AND A[k][2] = k*2 AND … AND A[k][k-1] = k * (k-1) (en verde)

b) A[1][k] = 1 + k AND A[2][k] = 2 + k AND … AND A[k-1][k] = (k-1) + k (en naranja) y

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (20%)

```
Función iEvaluacion ( A[1..n][1..n]: matriz de enteros; k : entero ) retorna ( b:booleano )
  caso
            k = 1 → VERDADERO
            k > 1 →  iEvaluacion ( A, k-1) AND AUX(A,k)
  fcaso
ffunción
```

La llamada inicial a la función iEvaluacion se realiza del siguiente modo: iEvaluacion(A, n).

La función AUX(A,k) es una función que calcula (1).-

```
Función AUX ( A[1..n][1..n]: matriz de enteros; k : entero ) retorna  ( b:booleano )
var i:entero, resul:booleano fvar
i = 0
resul = VERDADERO
mientras ( i < k-1 AND resul )
        i = i + 1;
        si ( A[ i ][ k ] ≠ i + k  OR A[ k ][ i ] ≠ k * i  ) entonces
             resul = FALSO
        fsi
fmientras
retorna resul
ffunción
```

2.- [2 puntos] Dada la siguiente función recursiva, se pide verificar formalmente su corrección:

$Q' \equiv \{ 1 \le k \le n+1 \wedge x \ge 0 \}$

Función iF2 (A[1..n]:vector de enteros, x:entero, k:entero) retorna (p:booleano)
caso

$k = n+1 \rightarrow$ VERDADERO

$k < n+1 \rightarrow$ iF2 (A, x, k+1) AND (A[k] = x^k)

fcaso
ffuncion
$R' \equiv \{ p = (\exists i)(A[i] = x^i : k \le i \le n) \}$

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

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

SOLUCIÓN:

1. $Q(\overline{x}) \Rightarrow B_t(\overline{x}) \vee B_{nt}(\overline{x})$

$1 \le k \le n+1 \wedge x \ge 0 \Rightarrow k = n+1 \vee k < n+1$ Se cumple.

2. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \Rightarrow Q(s(\overline{x}))$

$1 \le k \le n+1 \wedge x \ge 0 \wedge k < n+1 \Rightarrow k + 1 \le n+1$ Se cumple.

3. $Q(\overline{x}) \wedge B_t(\overline{x}) \Rightarrow R(\overline{x}, triv(\overline{x}))$

$1 \le k \le n+1 \wedge x \ge 0 \wedge k = n+1 \Rightarrow$ VERDADERO = $(\exists i)(A[i] = x^i : k \le i \le n)$

No se cumple, ya que cuando k=n+1 tendremos que el cuantificador ∃ afecta a un rango vacío, por lo que su valor es FALSO (elemento neutro de la operación OR). Sin embargo, la función iF2 devuelve VERDADERO. Para que se cumpla la postcondición en el caso trivial, la función debería retornar FALSO.

4. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \wedge R(s(\overline{x}), \overline{y}') \Rightarrow R(\overline{x}, c(\overline{y}', \overline{x}))$

$1 \le k \le n+1 \wedge x \ge 0 \wedge k < n+1 \wedge p' = (\exists i)(A[i] + x^i : k+1 \le i \le n) \Rightarrow p = p'$ AND $(A[k]= x^k)$

$p = p'$ AND $(A[k] = x^k) = (\exists i)(A[i] = x^i : k+1 \le i \le n)$ AND $(A[k] = x^k) \ne (\exists i)(A[i] = x^i : k \le i \le n)$

No se cumple, ya que la función de combinación de iF2 consiste en aplicar la operación AND al término k-ésimo cuando en realidad debería ser aplicar la operación OR al término k-ésimo, esto es,

$$p = p' \text{ OR } (A[k] = x^k)$$

ya que el cuantificador ∃ representa la operación OR, esto es,

$$(A[1] = x^1) \text{ OR } (A[2] = x^2) \text{ OR } \dots \text{OR } (A[k] = x^k)$$

5. Encontrar una función t:D → Z, tal que $Q(\overline{x}) \Rightarrow t(\overline{x}) \geq 0$

t(A,x,k) $\stackrel{\text{def}}{\equiv}$ n – k + 1

$1 \leq k \leq n+1 \wedge x \geq 0 \Rightarrow \quad n - k + 1 \geq 0$ Se cumple.

6. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \Rightarrow t(s(\overline{x})) < t(\overline{x})$

$1 \leq k \leq n+1 \wedge x \geq 0 \wedge k < n + 1 \Rightarrow \quad n - (k + 1) + 1 < n - k + 1$

$n - k < n - k + 1$ Se cumple.

Curso Académico 2016-2017

MODELO A

1.- [8 puntos] Dados dos vectores de enteros V[1..n] y W[1..n] con n>0, escribir una función recursiva que contabilice cuántas veces la suma de una componente de V con su correspondiente componente especular en W es igual a un entero x dado.

Ejemplos.

x = 10, V[1..3]={ 8, 6, 4 } y W[1..3]={ 6, 3, 2 } retorna 2 y

x = 10, V[1..4]={ 8, 6, 4, 3 } y W[1..4]={ 7, 2, 3, 1} retorna 1

Da respuesta a los siguientes apartados de manera clara, concisa y sin ambigüedad:

Nota.- Antes de mostrar la solución, resaltar que este problema guarda mucha similitud con el problema que consiste en indicar si un vector es capicúa o no, problema resuelto en clase y recogido en las diapositivas del tema "Diseño de Algoritmos Recursivos".

a) Especificación formal de la función (15%)

$Q \equiv \{ n > 0 \}$

Función Evaluacion (V[1..n], W[1..n]:vectores de enteros, x:entero) retorna (b:entero)

$R \equiv \{ b = (N\ k)(V[k] + W[n-k+1] = x : 1 \leq k \leq n)\}$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada vector le asociaremos un natural n que corresponde al número de elementos del vector. Seguidamente aplicaremos inducción fuerte sobre dicho natural n.

c) Realizar una inmersión no final, escribir la especificación formal de la función inmersora ye indicar cómo ha de efectuarse la primera llamada a la función inmersora (10%)

Sustituimos 1 por i y n por j en *R*:

$Q' \equiv \{ 1 \leq i \leq j \leq n \}$

Función iEvaluacion (V[1..n], W[1..n]:vector de enteros, x, i, j:entero) retorna (b:entero)

$R' \equiv \{ b = (N\ k)(V[k] + W[j-k+i] = x : i \leq k \leq j)\}$

Primera llamada a la función: Evaluacion(V,W,x) = iEvaluacion(V,W,x,1,n)

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(V,W,x,i,j), para pasar al problema sucesor eliminaremos una componente al comienzo de la sección y otra al final. Así, el subproblema sucesor será iEvaluacion (V,W,x,i+1,j-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

$B_t \equiv$ i = j la sección del vector a tratar tiene un único elemento.

$B_t \equiv$ i+1 = j la sección del vector a tratar tiene dos elementos.

$B_{nt} \equiv$ i+1 < j la sección del vector a tratar tiene más de dos elementos.

f) **Explicar razonadamente** la solución propuesta para el caso trivial, esto es, *triv*($\bar{x}$) (15%)

En el caso de que la sección tratada de los vectores tuviera un elemento cada una, habría que comprobar si dichos elementos sumados coinciden con el valor x. Si así fuera, la función devolvería 1 y sino retornaría 0.

En el caso de que la sección tratada de los vectores tuviera dos elementos cada una, habría que comprobar si la suma de la primera componente de V con la última de W es igual a x y de igual modo si la suma de la primera componente de W con la última de V es igual a x. En el caso de que una comprobación fuera cierta, su valor asociado sería 1, por lo que si ambas comprobaciones fuesen ciertas la función retornaría 2. Si sólo lo fuese una de ellas, la función retornaría 1; mientras que si no lo fuera ninguna, la función retornará 0.

g) **Explicar razonadamente** la solución propuesta para el caso no trivial, esto es, explicar la función de combinación *c*($\bar{x}$, $\bar{y}$´) (25%)

Llamaremos b a la solución del problema iEvaluacion(V,W,x,i,j) el cual consiste en determinar cuántas veces la suma de una componente de V[i..j] con su correspondiente componente especular en W[i..j] es igual a un entero x.

Hipótesis de recurrencia:
Supongamos conocida la solución del subproblema iEvaluacion(V,W,x,i+1,j-1). Llamaremos a dicho valor, b´. Esto quiere decir que suponemos conocido cuántas veces la suma de una componente de V[i+1..j-1] con su correspondiente componente especular en W[i+1..j-1] es igual a un entero x.

Para obtener la solución del problema iEvaluacion(V,W,x,i,j) a partir de la solución del subproblema iEvaluacion(V,W,x,i+1,j-1) bastaría comprobar:

(V [i] + W [j] = x) y (V [j] + W [i] = x)

Si ambas comprobaciones fuesen ciertas implicaría sumar 2 a b’ para obtener b. Si sólo una de las comprobaciones fuese cierta habría que sumar 1 a b’ para obtener b. Si no lo fuese ninguna, b=b’.

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (10%)

```
Función iEvaluacion ( V[1..n], W[1..n]:vector de enteros, x, i, j:entero) retorna ( b:entero )
  caso
        i = j → ( V [ i ] + W [ j ] = x )
        i + 1 = j → ( V [ i ] + W [ j ] = x ) + ( V [ j ] + W [ i ] = x )
        i +1 < j → iEvaluacion ( V, W, x, i+1, j-1) + ( V [ i ] + W [ j ] = x ) + ( V [ j ] + W [ i ] = x )
  fcaso
ffunción
```

i) Demostrar los pasos 3, 4 y 5 de verificación formal para el algoritmo obtenido en el apartado anterior (10%)

$3.\ Q(\overline{x}) \wedge B_t(\overline{x}) \Rightarrow R(\overline{x}, triv(\overline{x}))$

1 ≤ i ≤ j ≤ n ∧ i = j ⇒ b = (Nk)(V[k]+W[j – k + i] = x : i ≤ k ≤ j

Si i=j eso quiere decir que b = (Nk)(V[k]+W[j – k + i] = x : i ≤ k ≤ i, por lo que

b = (V[i] + W[j – i + i] = x)

b = (V[i] + W[j] = x)

que es lo que devuelve la función para el caso trivial i=j (c.q.d.)

Si i+1=j eso quiere decir que b = (Nk)(V[k]+W[j – k + i] = x : i ≤ k ≤ i+1, por lo que

b = (V[i] + W[j – i + i] = x) + (V[i+1] + W[j – (i+1) + i] = x)

b = (V[i] + W[j] = x) + (V[i+1] + W[j - 1] = x)

al ser i +1=j y por tanto i=j-1 sustituimos y quedaría

b = (V[i] + W[j] = x) + (V[j] + W[i] = x)

que es lo que devuelve la función para el caso trivial i+1=j (c.q.d.)

$4.\ Q(\overline{x}) \wedge B_{nt}(\overline{x}) \wedge R(s(\overline{x}), \overline{y}`) \Rightarrow R(\overline{x}, c(\overline{y}`, \overline{x}))$

1 ≤ i ≤ j ≤ n ∧ i+1 < j ∧ b' = (Nk)(V[k]+W[j – k + i] = x : i+1 ≤ k ≤ j-1) ⇒
b=(Nk)(V[k]+W[j–k + i] = x : i ≤ k ≤ j)

Veamos,

b = b' + (V [i] + W [j] = x) + (V [j] + W [i] = x) =

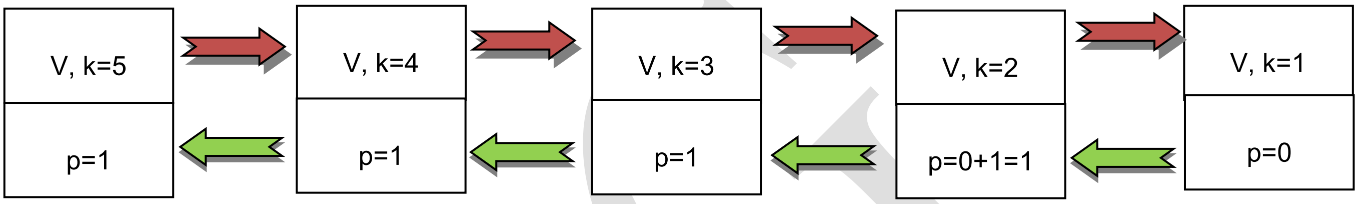
= (Nk)(V[k]+W[j – k + i] = x : i+1 ≤ k ≤ j-1) + (V [i] + W [j] = x) + (V [j] + W [i] = x) =

= (Nk)(V[k]+W[j – k + i] = x : i ≤ k ≤ j) (c.q.d.)

2.- [1 punto] Dada la siguiente función recursiva:

```
Función iF2 (V[1..n]:vector de enteros, k:entero) retorna (p:entero)
var p:entero fvar;
si ( k = 1 ) entonces retorna 0
sino        p = iF2( V, k-1 )
            si ( V [ k ] %2 = 0 y k %2 = 0 ) entonces p = p + 1 fsi
            retorna p
fsi
ffuncion
```

Dibuja la lista de llamadas que se realiza para V[1..5] = { 1, 2, 6, 7, 9 } y k = 5 mostrando los valores de los parámetros de cada llamada, así como los resultados que se obtienen para cada una de ellas.



El resultado que finalmente proporciona la función es 1, esto es, iF2(V,5)=1.

3.- [1 punto] Responde a las siguientes cuestiones de manera clara, concisa y sin ambigüedad:

a) Si hacemos una recursión sobre un vector V[1..n] con n ≥ 1, en la que "se elimina" la primera componente del vector para pasar al problema sucesor, ¿cuál/es será/n el/los caso/s trivial/es?
Al aplicar el proceso de inmersión sustituiremos 1 por j. De ese modo el caso trivial correspondería a j=n, esto es, cuando la sección a tratar es de tamaño 1.

b) Si hacemos una recursión sobre un número n, donde n≥0, en la que hay tres problemas sucesores, n-1, n-2 y n-3, ¿cuál/es será/n el/los caso/s trivial/es?
Los casos triviales corresponden con n < 3, esto es, n=0, n=1 y n=2.

c) Si hacemos una recursión sobre una matriz M[1..n][1..m] con n ≥ 0 y m ≥ 0, en la que "se eliminan" la primera y la última fila de la matriz para pasar al problema sucesor, ¿cuál/es será/n el/los caso/s trivial/es?
Al aplicar el proceso de inmersión sustituiremos 1 por i y n por j. De ese modo, el caso trivial correspondería a i > j, esto es, cuando la sección de filas a tratar es vacía.

MODELO B

1.- [8 puntos] Dados dos vectores de enteros V[1..n] y W[1..n] con n>0, escribir una función recursiva que determine si el vector W es la imagen especular del vector V, o no.

Ejemplos.

V[1..3]={ 8, 6, 4 } y W[1..3]={ 4, 6, 8 } retorna VERDADERO y

V[1..4]={ 8, 6, 4, 3 } y W[1..4]={ 7, 4, 6, 8} retorna FALSO

Da respuesta a los siguientes apartados de manera clara, concisa y sin ambigüedad:

Nota.- Antes de mostrar la solución, resaltar que este problema guarda mucha similitud con el problema que consiste en indicar si un vector es capicúa o no, problema resuelto en clase y recogido en las diapositivas del tema "Diseño de Algoritmos Recursivos".

a) Especificación formal de la función (15%)

$Q \equiv \{ n > 0 \}$

Función Evaluacion (V[1..n], W[1..n]:vectores de enteros) retorna (b:booleano)

$R \equiv \{ b = (\forall k)(V[k] = W[n-k+1] : 1 \leq k \leq n)\}$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada vector le asociaremos un natural n que corresponde al número de elementos del vector. Seguidamente aplicaremos inducción fuerte sobre dicho natural n.

c) Realizar una inmersión no final, escribir la especificación formal de la función inmersora ye indicar cómo ha de efectuarse la primera llamada a la función inmersora (10%)

Sustituimos 1 por i y n por j en *R*:

$Q' \equiv \{ 1 \leq i \leq j \leq n \}$

Función iEvaluacion (V[1..n], W[1..n]:vector de enteros, i, j:entero) retorna (b:booleano)

$R' \equiv \{ b = (\forall k)(V[k] = W[j-k+i] : i \leq k \leq j)\}$

Primera llamada a la función: Evaluacion(V,W,x) = iEvaluacion(V,W,x,1,n)

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(V,W,i,j), para pasar al problema sucesor eliminaremos una componente al comienzo de la sección y otra al final de la sección a tratar. Así, el subproblema sucesor será iEvaluacion (V,W,i+1,j-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

$B_t \equiv$ i = j la sección del vector a tratar tiene un único elemento.

$B_t \equiv$ i+1 = j la sección del vector a tratar tiene dos elementos.

$B_{nt} \equiv$ i+1 < j la sección del vector a tratar tiene más de dos elementos.

f) **Explicar razonadamente** la solución propuesta para el caso trivial, esto es, *triv*($\bar{x}$) (15%)

En el caso de que la sección tratada de los vectores tuviera un elemento cada una, habría que comprobar si dichos elementos coinciden. Si así fuera, la función devolvería VERDADERO y sino retornaría FALSO.

En el caso de que la sección tratada de los vectores tuviera dos elementos cada una habría que comprobar si la primera componente de V coincide con la última de W y de igual modo si la primera componente de W coincide con la última de V. Por lo que si ambas comprobaciones fuesen ciertas la función retornaría VERDADERO; en caso contrario, FALSO.

g) **Explicar razonadamente** la solución propuesta para el caso no trivial, esto es, explicar la función de combinación $c(\bar{x}, \bar{y}')$ (25%)

Llamaremos b a la solución del problema iEvaluacion(V,W,x,i,j) el cual consiste en determinar si W[i..j] es la imagen especular de V[i..j].

Hipótesis de recurrencia:
Supongamos conocida la solución del subproblema iEvaluacion(V,W,i+1,j-1). Llamaremos a dicho valor, b´. Esto quiere decir que suponemos conocido si W[i+1..j-1] es la imagen especular de V[i+1..j-1].

Para obtener la solución del problema iEvaluacion(V,W,i,j) a partir de la solución del subproblema iEvaluacion(V,W,i+1,j-1) bastaría comprobar:

(V [i] = W [j]) y (V [j] = W [i])

Si ambas comprobaciones fuesen ciertas, W[i..j] es la imagen especular de V[i..j] siempre cuando W[i+1..j-1] fuera la imagen especular de V[i+1..j-1], esto es,

b = b ' AND (V [i] = W [j]) AND (V [j] = W [i])

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (10%)

```
Función iEvaluacion ( V[1..n], W[1..n]:vector de enteros, i, j:entero) retorna ( b:booleano )
  caso
        i = j → ( V [ i ] = W [ j ])
        i + 1 = j → ( V [ i ] = W [ j ] ) AND ( V [ j ] = W [ i ] )
        i +1 < j → iEvaluacion ( V, W, i+1, j-1) AND ( V [ i ] = W [ j ]) AND ( V [ j ] = W [ i ] )
  fcaso
ffunción
```

i) Demostrar los pasos 3, 4 y 5 de verificación formal para el algoritmo obtenido en el apartado anterior (10%)

3. $Q(\overline{x}) \wedge B_t(\overline{x}) \Rightarrow R(\overline{x}, triv(\overline{x}))$

$1 \le i \le j \le n \wedge i = j \Rightarrow b = (\forall k)(V[k]=W[j - k + i] : i \le k \le j$

Si i=j eso quiere decir que $b = (\forall k)(V[k]=W[j - k + i] : i \le k \le i$, por lo que

$$b = (V[i] = W[j - i + i])$$

$$b = (V[i] = W[j])$$

que es lo que devuelve la función para el caso trivial i=j (c.q.d)

Si i+1=j eso quiere decir que $b = (\forall k)(V[k]=W[j - k + i] : i \le k \le i+1$, por lo que

$$b = (V[i] = W[j - i + i]) \text{ AND } (V[i+1] = W[j - (i+1) + i])$$

$$b = (V[i] = W[j]) \text{ AND } (V[i+1] = W[j - 1])$$

al ser i +1=j y por tanto i=j-1 sustituimos y quedaría

$$b = (V[i] = W[j]) \text{ AND } (V[j] = W[i])$$

que es lo que devuelve la función para el caso trivial i+1=j (c.q.d)

4. $Q(\overline{x}) \wedge B_{nt}(\overline{x}) \wedge R(s(\overline{x}), \overline{y}`) \Rightarrow R(\overline{x}, c(\overline{y}`, \overline{x}))$

$1 \le i \le j \le n \wedge i +1 < j \wedge b' = (\forall k)(V[k]=W[j - k + i]: i+1 \le k \le j-1) \Rightarrow$
$b = (\forall k)(V[k]=W[j - k + i]: i \le k \le j)$

Veamos,

$b = b' \text{ AND } (V[i] = W[j]) \text{ AND } (V[j] = W[i]) =$

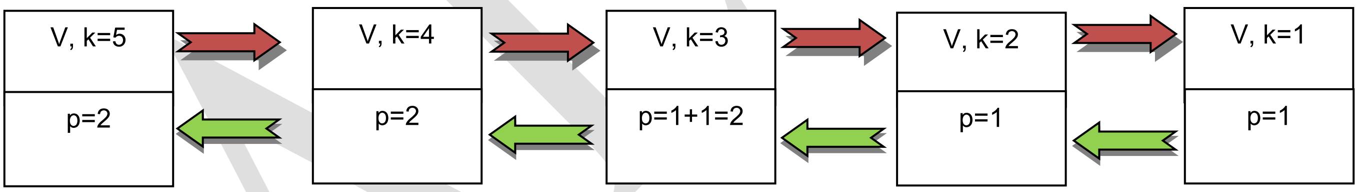
$= (\forall k)(V[k]=W[j-k+i] : i+1 \leq k \leq j-1) \text{ AND } (V[i] = W[j]) \text{ AND } (V[j] = W[i]) =$

$= (\forall k)(V[k]+W[j-k+i] = x : i \leq k \leq j)$ (c.q.d)

2.- [1 punto] Dada la siguiente función recursiva:

```
Función iF2 (V[1..n]:vector de enteros, k:entero) retorna (p:entero)
var p:entero fvar;
si ( k = 1 ) entonces
                        si ( V [ k ] %2 ≠ 0 ) entonces retorna 1
                           sino retorna 0
                          fsi
sino          p = iF2( V, k-1 )
              si ( V [ k ] %2 ≠ 0 y k %2 ≠ 0 ) entonces p = p + 1 fsi
              retorna p
fsi
ffuncion
```

Dibuja la lista de llamadas que se realiza para V[1..5] = { 7, 6, 3, 8, 2 } y k = 5 mostrando los valores de los parámetros de cada llamada, así como los resultados que se obtienen para cada una de ellas.



El resultado que finalmente proporciona la función es 2, esto es, iF2(V,5)=2.

3.- [1 punto] Responde a las siguientes cuestiones de manera clara, concisa y sin ambigüedad:

a) Si hacemos una recursión sobre un vector V[1..n] con $n \geq 1$, en la que "se elimina" la primera componente del vector para pasar al problema sucesor, ¿cuál/es será/n el/los caso/s o trivial/es?
Al aplicar el proceso de inmersión sustituiremos 1 por j. De ese modo el caso trivial correspondería a j=n, esto es, cuando la sección a tratar es de tamaño 1.

b) Si hacemos una recursión sobre un número n, donde $n \geq 0$, en la que hay tres problemas sucesores, n-1, n-2 y n-3, ¿cuál/es será/n el/los caso/s o trivial/es?
Los casos triviales corresponden con $n < 3$, esto es, n=0, n=1 y n=2.

c) Si hacemos una recursión sobre una matriz M[1..n][1..m] con $n \geq 0$ y $m \geq 0$, en la que "se eliminan" la primera y la última columna de la matriz para pasar al problema sucesor, ¿cuál/es será/n el/los caso/s o trivial/es?
Al aplicar el proceso de inmersión sustituiremos 1 por i y n por j. De ese modo el caso trivial correspondería a $i > j$, esto es, cuando la sección de columnas a tratar es vacía.

Curso Académico 2015-2016

MODELO A

1.- [8 puntos] Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

"Dada una matriz A[1..n][1..n] de enteros, siendo n ≥ 1, diseñar una función recursiva que determine si existen dos elementos consecutivos de la diagonal principal cuyo producto sea igual a un valor entero v."

Ejemplos:

$A[1..3][1..3] = \begin{bmatrix} 0 & 1 & 2 \\ 4 & 5 & 2 \\ 1 & 3 & 8 \end{bmatrix}$ v = 0 retornará VERDADERO

$A[1..3][1..3] = \begin{bmatrix} 0 & 1 & 2 \\ 4 & 5 & 4 \\ 1 & 3 & 10 \end{bmatrix}$ v = 40 retornará FALSO

Da respuesta a los siguientes apartados de manera clara, concisa y sin ambigüedad:

a) Especificación formal de la función (15%)

$Q \equiv \{ n \geq 1 \}$

Función Evaluacion (A[1..n][1..n]:matriz de enteros, v:entero) retorna (b:booleano)

$R \equiv \{ b = (\exists i)(A[i-1][i-1] * A[i][i] = v : 2 \leq i \leq n) \}$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada matriz le asociaremos un natural n que corresponde al orden de la matriz. Seguidamente aplicaremos inducción débil sobre dicho natural n.

c) Realizar una inmersión no final y escribir la especificación formal de la función inmersora (10%)

Sustituimos n por j en *R*:

$Q' \equiv \{ \ 1 \leq j \leq n \ \}$

Función iEvaluacion (A[1..n][1..n]:matriz de enteros, v:entero, j:entero) retorna (b:booleano)

$R' \equiv \{ b = (\exists i)(A[i-1][i-1] * A[i][i] = v : 2 \leq i \leq j) \}$

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(A, v, j), para pasar al siguiente problema eliminamos una fila y una columna. Así, el subproblema sucesor será iEvaluacion(A, v, j-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

$B_t \equiv$ j = 1 la sección de la matriz tiene un única fila y columna, esto es, un único elemento.

$B_{nt} \equiv$ j > 1 la sección de la matriz tiene varias filas y columnas, el mismo número de ambas.

f) **Explicar razonadamente** la solución propuesta para el caso trivial, esto es, *triv*($\bar{x}$) (15%)

En el caso de que la sección de la matriz tratada tuviera un único elemento (j = 1), la solución del problema sería FALSO ya que no existen dos elementos consecutivos en la diagonal cuyo producto pudiera ser v.

g) **Explicar razonadamente** la solución propuesta para el caso no trivial, esto es, explicar la función de combinación $c(\bar{x}, \overline{y'})$ (25%)

La solución del problema iEvaluacion(A, v, j) consiste en determinar si existen dos elementos consecutivos de la diagonal, A[k][k] con k=1, 2.., j, cuyo producto sea v, esto es,

$$(\exists\ i)(\ A[i-1][i-1] * A[i][i] = v: 2 \le i \le j\ ),$$

llamamos a dicho resultado, b.

Hipótesis de recurrencia: Supongamos conocida la solución del subproblema iEvaluacion(A, v,j-1). Llamaremos a dicho valor, b´. Esto quiere decir que suponemos conocido si existen dos elementos consecutivos de la diagonal, A[k][k] con k=1, 2.., j-1, cuyo producto sea v, esto es,

$$(\exists\ i)(\ A[i-1][i-1] * A[i][i] = v: 2 \le i \le j-1\ )$$

Para obtener la solución del problema iEvaluacion(A, v, j) a partir de la solución del subproblema iEvaluacion(A,v,j-1) bastaría comprobar si A[j-1][j-1] * A[j][j] es igual v.

$$(\exists\ i)(\ A[i-1][i-1] * A[i][i] = v: 2 \le i \le j-1\ )$$

Por tanto,

$$b = b´ \text{ OR } (\ A[j-1][j-1] * A[j][j] = v)$$

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (10%)

```
Función iEvaluacion ( A[1..n][1..n]: matriz de enteros; v:entero, j : entero ) retorna  ( b:booleano )
  caso
     j = 1 → FALSO
     j > 1 →  iEvaluacion ( A, v, j-1) OR ( A[j-1][j-1] * A[j][j] = v)
  fcaso
ffunción
```

donde A[j-1][j-1] * A[j][j] = v, proporciona el valor VERDADERO si A[j-1][j-1] * A[j][j] es igual a v y FALSO, en caso contrario.

La llamada inicial a la función iEvaluacion se realiza del siguiente modo: iEvaluacion(A, v, n)

i) Demostrar los pasos 3, 4 y 5 de verificación formal para el algoritmo obtenido en el apartado anterior (10%)

$3. Q(\overline{x}) \wedge B_t(\overline{x}) \Rightarrow R(\overline{x}, triv(\overline{x}))$

$1 \le j \le n \wedge j = 1 \Rightarrow b = (\exists i)(A][i-1][i-1] * A[i][i] = v: 2 \le i \le j)$

¿ $b = (\exists i)(A][i-1][i-1] * A[i][i] = v: 2 \le i \le j)$ cuando j = 1?

Cuando j=1, el predicado $(\exists i)(A][i-1][i-1] * A[i][i] = v: 2 \le i \le j)$ es

$$(\exists i)(A][i-1][i-1] * A[i][i] = v: 2 \le i \le 1)$$

donde ∃ actúa sobre un rango vacío, siendo su valor por definición, FALSO.

Dado que la función del apartado h) retorna FALSO, se cumple la base de la recurrencia.

$4. Q(\overline{x}) \wedge B_{nt}(\overline{x}) \wedge R(s(\overline{x}), \overline{y}`) \Rightarrow R(\overline{x}, c(\overline{y}`, \overline{x}))$

Tendremos que demostrar que si ocurre:

$1 \le j \le n \wedge j > 1 \wedge b' = (\exists i)(A][i-1][i-1] * A[i][i] = v: 2 \le i \le j-1)$

entonces

b = b' OR (A[j-1][j-1] * A[j][j] = v)

nos proporciona la solución correcta para el problema iEvaluacion(A, j).

Veámoslo:

En la siguiente expresión b = b' OR (A[j-1][j-1] * A[j][j] = v)

sustituiremos b' por su valor (hipótesis de inducción), esto es,

b = b' OR (A[j-1][j-1] * A[j][j] = v) =

= $(\exists i)(A][i-1][i-1] * A[i][i] = v: 2 \le i \le j-1)$ OR (A[j-1][j-1] * A[j][j] = v) =

= $(\exists i)(A][i-1][i-1] * A[i][i] = v: 2 \le i \le j)$

Como podemos ver, esta expresión corresponde con la postcondición, para el problema iEvaluacion(A,j); por tanto, se cumple.

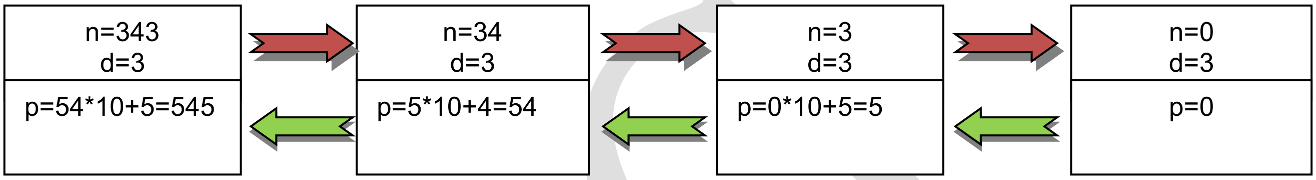
5. Definimos la función t del siguiente modo: $t (A, v, j) \stackrel{\text{def}}{=} j$ por tanto

$$1 \le j \le n \Rightarrow j \ge 0$$

2.- [1 punto] Dada la siguiente función recursiva:

Q ≡ { n ≥ 0 ∧ 0 ≤ d ≤ 9 }
Funcion F2 (n:entero, d:entero) retorna (p:entero)
  si n = 0 entonces retorna 0
  sino si n % 10 = d retorna F2(n/10, d) * 10 + 5
        sino retorna F2(n/10,d) * 10 + n % 10
        fsi
  fsi
ffuncion

Dibuja la lista de llamadas que se realiza para la llamada F2(343,3), mostrando los valores de los parámetros de cada llamada, así como los resultados que se obtienen para cada una de ellas.



El resultado que finalmente proporciona la función es 545, esto es, F2(343,3)=545

3.- [1 punto] Responde a las siguientes cuestiones de manera clara, concisa y sin ambigüedad:

a) Si hacemos una recursión sobre un vector V[1..n] con n ≥ 1, en la que se eliminan la primera y la última componente del vector para pasar al problema sucesor, ¿cuál/es será/n el/los caso/s o trivial/es?

Al aplicar el proceso de inmersión sustituiremos 1 por i y n por j. De ese modo la detección de los casos triviales correspondería a i=j y i=j-1:

- i=j corresponde a una sección de vector de tamaño 1
- i=j-1 corresponde a una sección de vector de tamaño 2

b) Si hacemos una recursión sobre un número n, donde n≥0, en la que hay dos problemas sucesores, n-1 y n-3, ¿cuál/es será/n el/los caso/s o trivial/es?

Los casos triviales corresponden con n < 3, esto es, n=0, n=1 y n=2.

MODELO B

1.- [8 puntos] Se quiere diseñar un algoritmo recursivo que resuelva el siguiente problema:

"Dada una matriz A[1..n][1..n] de enteros, siendo n ≥ 1, diseñar una función recursiva que contabilice el número de veces que la suma de dos elementos consecutivos de la diagonal principal sea igual a un valor entero v."

Ejemplos:

$\mathrm{A}[1..3][1..3] = \begin{bmatrix} 1 & 1 & 2 \\ 4 & 6 & 2 \\ 1 & 3 & 1 \end{bmatrix}$ v = 7 retornará 2

$\mathrm{A}[1..3][1..3] = \begin{bmatrix} 0 & 2 & 2 \\ 4 & 5 & 2 \\ 1 & 3 & 10 \end{bmatrix}$ v = 7 retornará 0

Da respuesta a los siguientes apartados de manera clara, concisa y sin ambigüedad:

a) Especificación formal de la función (15%)

$Q \equiv \{\, n \geq 1 \,\}$

Función Evaluacion (A[1..n][1..n]:matriz de enteros, v:entero) retorna (b:entero)

$R \equiv \{\, b = (N\ i)(A[i-1][i-1] + A[i][i] = v : 2 \leq i \leq n)\}$

b) Describir el principio de inducción utilizado (5%)

Inducción Noetheriana: A cada matriz le asociaremos un natural n que corresponde al orden de la matriz. Seguidamente aplicaremos inducción débil sobre dicho natural n.

c) Realizar una inmersión no final y escribir la especificación formal de la función inmersora (10%)

Sustituimos n por j en *R*:

$Q' \equiv \{\ 1 \leq j \leq n\ \}$

Función iEvaluacion (A[1..n][1..n]:matriz de enteros, v:entero, j:entero) retorna (b:entero)

$R' \equiv \{\, b = (Ni)(A[i-1][i-1] + A[i][i] = v : 2 \leq i \leq j)\}$

d) Identificar el sucesor, o sucesores, de un problema dado (5%)

Dado el problema iEvaluacion(A, v, j), para pasar al siguiente eliminamos una fila y una columna. Así, el subproblema sucesor será iEvaluacion(A, v, j-1).

e) Establecer las condiciones de caso/s trivial/es y caso/s no trivial/es (5%)

$B_t \equiv\ j = 1$ la sección de la matriz tiene un única fila y columna, esto es, un único elemento.

$B_{nt} \equiv\ j > 1$ la sección de la matriz tiene varias filas y columnas, el mismo número de ambas.

f) Explicar razonadamente la solución propuesta para el caso trivial, esto es, *triv*($\bar{x}$) (15%)

En el caso de que la sección de la matriz tratada tuviera un único elemento (j = 1), la solución del problema sería 0 ya que no existen dos elementos consecutivos en la diagonal.

g) Explicar razonadamente la solución propuesta para el caso no trivial, esto es, explicar la función de combinación *c*($\bar{x}$, $\overline{y}'$) (25%)

La solución del problema iEvaluacion(A, v, j) consiste en determinar el número de veces que la suma de dos elementos consecutivos de la diagonal principal, A[k][k] con k=1, 2.., j, sea igual a un valor entero v, esto es,

(N i)(A][i-1][i-1] + A[i][i] = v: 2 ≤ i ≤ j),

llamamos a dicho resultado, b.

Hipótesis de recurrencia: Supongamos conocida la solución del subproblema iEvaluacion(A, v,j-1). Llamaremos a dicho valor, b´. Esto quiere decir que suponemos conocido el número de veces que la suma de dos elementos consecutivos de la diagonal principal, A[k][k] con k=1, 2.., j-1, sea igual a un valor entero v, esto es,

(N i)(A][i-1][i-1] * A[i][i] = v: 2 ≤ i ≤ j-1)

Para obtener la solución del problema iEvaluacion(A, v, j) a partir de la solución del subproblema iEvaluacion(A,v,j-1) bastaría comprobar si A[j-1][j-1] + A[j][j] es igual v y si así fuera sumar 1 a b´.

Por tanto,

b = b´ + (A[j-1][j-1] + A[j][j] = v)

h) Escribir la función inmersora en pseudocódigo, incluyendo función/es auxiliar/es si se precisan (10%)

```
Función iEvaluacion ( A[1..n][1..n]: matriz de enteros; v:entero, j : entero ) retorna  ( b:entero )
  caso
     j = 1 → 0
     j > 1 →  iEvaluacion ( A, v, j-1) + ( A[j-1][j-1] + A[j][j] = v)
  fcaso
ffunción
```

donde A[j-1][j-1] + A[j][j] = v, proporciona el valor 1 si A[j-1][j-1] + A[j][j] es igual a v y 0, en caso contrario.

La llamada inicial a la función iEvaluacion se realiza del siguiente modo: iEvaluacion(A, v, n)

i) Demostrar los pasos 3, 4 y 5 de verificación formal para el algoritmo obtenido en el apartado anterior (10%)

$3. Q(\overline{x}) \wedge B_t(\overline{x}) \Rightarrow R(\overline{x}, triv(\overline{x}))$

1 ≤ j ≤ n ∧ j = 1 ⇒ b = (N i)(A][i-1][i-1] + A[i][i] = v: 2 ≤ i ≤ j)
¿ b = (N i)(A][i-1][i-1] + A[i][i] = v: 2 ≤ i ≤ j) cuando j = 1?

Cuando j=1, el predicado (N i)(A][i-1][i-1] + A[i][i] = v: 2 ≤ i ≤ j) es

(N i)(A][i-1][i-1] + A[i][i] = v: 2 ≤ i ≤ 1)

donde N actúa sobre un rango vacío, siendo su valor por definición, 0.

Dado que la función del apartado h) retorna 0 , se cumple la base de la recurrencia.

$4. Q(\overline{x}) \wedge B_{nt}(\overline{x}) \wedge R(s(\overline{x}), \overline{y}`) \Rightarrow R(\overline{x}, c(\overline{y}`, \overline{x}))$

Tendremos que demostrar que si ocurre:

1 ≤ j ≤ n ∧ j > 1 ∧ b' = (N i)(A][i-1][i-1] + A[i][i] = v: 2 ≤ i ≤ j-1)

entonces

b = b' + (A[j-1][j-1] + A[j][j] = v)

nos proporciona la solución correcta para el problema iEvaluacion(A, j).

Veámoslo:

En la siguiente expresión b = b' + (A[j-1][j-1] + A[j][j] = v)

sustituiremos b' por su valor (hipótesis de inducción), esto es,

b = b' + (A[j-1][j-1] + A[j][j] = v) =

= (N i)(A][i-1][i-1] + A[i][i] = v: 2 ≤ i ≤ j-1) + (A[j-1][j-1] + A[j][j] = v) =

= (N i)(A][i-1][i-1] + A[i][i] = v: 2 ≤ i ≤ j)

Como podemos ver, esta expresión corresponde con la postcondición, para el problema iEvaluacion(A,j); por tanto, se cumple.

5. Definimos la función t del siguiente modo: t (A, v, j) $\stackrel{\text{def}}{=}$ j por tanto

1 ≤ j ≤ n ⇒ j ≥ 0

2.- [1 punto] Responde a las siguientes cuestiones de manera clara, concisa y sin ambigüedad:

a) Si hacemos una recursión sobre un vector V[1..n] con $n \geq 1$, en la que se eliminan la primera y la última componente del vector para pasar al problema sucesor, ¿cuál/es será/n el/los caso/s o trivial/es?

Al aplicar el proceso de inmersión sustituiremos 1 por i y n por j. De ese modo la detección de los casos triviales correspondería a i=j y i=j-1:

- i=j corresponde a una sección de vector de tamaño 1
- i=j-1 corresponde a una sección de vector de tamaño 2

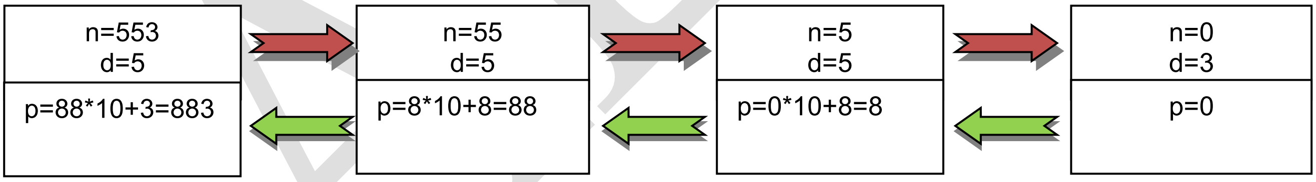
b) Si hacemos una recursión sobre un número n, donde $n \geq 0$, en la que hay dos problemas sucesores, n-1 y n-3, ¿cuál/es será/n el/los caso/s o trivial/es?

Los casos triviales corresponden con $n < 3$, esto es, n=0, n=1 y n=2.

3.- [1 punto] Dada la siguiente función recursiva:

```
Q ≡ { n ≥ 0 ∧ 0 ≤ d ≤ 9 }
Funcion F3 ( n:entero, d:entero  ) retorna ( p:entero )
  si  n = 0 entonces retorna 0
  sino   si  n % 10 = d  retorna F3(n/10, d) * 10 + 8
         sino retorna F3(n/10,d) * 10  +  n % 10
         fsi
  fsi
ffuncion
```

Dibuja la lista de llamadas que se realiza para la llamada F3(553,5), mostrando los valores de los parámetros de cada llamada, así como los resultados que se obtienen para cada una de ellas.



El resultado que finalmente proporciona la función es 883, esto es, F3(553,5)=883