FE DE ERRATAS (**1ª Reimpresión 2016**)

Pág	Errata
Pág. 24	En el segundo párrafo, donde dice: ...2 componentes conexas: una formada por los nodos {1,2,3,4,5} y otra por los nodos {6,7} debe decir: ...2 componentes conexas: una formada por los nodos {1,2,3,4,5,6} y otra por los nodos {6,7}
Pág. 35	En la función Hundir hay que suprimir la declaración local `i: natural;`
Pág. 40	Donde dice $\sum_{i=1}^{k} 2^{i-1} = 2^{k-1}$ Debe decir $\sum_{i=1}^{k} 2^{i-1} = 2^{k} - 1$
Pág. 40	Donde dice: “suponiendo sin pérdida de generalidad que n es potencia de 2, es decir que el árbol es binario completo” debe decir “suponiendo sin pérdida de generalidad que n es potencia de 2”
Pág 43	Donde dice “Un montículo binomial con n elementos consiste **en al menos** log n + 1 árboles binomiales, ...” debe decir

	"Un montículo binomial con n elementos consiste **como mucho en** log n + 1 árboles binomiales, ..."
Pág. 54	Al final de la sección 2.3, en el último párrafo donde pone "Apicando" debe decir "Aplicando".
Pág. 62	Cuando se define el conjunto finito de tipos de moneda T= $\{m^0, m^1, \ldots, m^n\}$ en el párrafo tercero y cuarto de la página, la definición debería ser: T= $\{m^0, m^1, \ldots, m^{n-1}\}$.
Pág. 66	En el primer párrafo pone: "Si el grafo tiene algunas aristas de peso 0 también podrá haber más de una solución con igual peso, pero distinto número de aristas. En este caso se trataría de seleccionar la solución con menor número de aristas. Nótese que un grafo conexo con n nodos debe tener al menos n-1 aristas ..." y debería poner: "Si el grafo tiene algunas aristas de peso 0 también podrá haber más de una solución con igual peso, pero distintas aristas. Nótese que un grafo conexo con n nodos debe tener al menos n-1 aristas ..."
Pág. 70	En el apartado "Coste" dice: "El bucle principal se ejecuta n-1 veces y dentro de este bucle hay una secuencia de 2 bucles que se ejecutan n-2 veces, por lo que ..." y debería poner: "El bucle principal se ejecuta n-1 veces y dentro de este bucle hay una secuencia de 2 bucles que se ejecutan **n-1** veces, por lo que ..."
Pág. 118	La función Combinar queda como sigue: **fun** Combinar(a,b:**entero**;i,j:**natural;**v:**vector** [1..n] de **natural**):**entero** **si** a= -1 ∧ b= -1 **entonces dev** -1 **fsi** **si** a= -1 ∧ b ≠ -1 **entonces dev** ComprobarMayoritario(i,j,b,v) **fsi** **si** a ≠ -1 ∧ b = -1 **entonces dev** ComprobarMayoritario(i,j,a,v) **fsi** **si** a ≠ -1 ∧ b ≠ -1 **entonces** **si** a = b **entonces dev** a **fsi** **sino si** ComprobarMayoritario(i,j,a,v) = a **entonces dev** a **fsi** **sino si** ComprobarMayoritario(i,j,b,v) = b **entonces dev** b **fsi** **sino dev** -1 **fsi** **fsi** **ffun**

Pág. 118	La función ComprobarMayoritario debe parametrizarse y queda como sigue: **fun** ComprobarMayoritario (i,j:**natural**;x:**entero;**v:**vector** [1..n] **de natural**):**entero** **var** c:**natural** **fvar** c←0 **para** k←i **hasta** j **hacer** **si** v[k]=x **entonces** c←c+1 **fsi** **fpara** **si** c > (j-i+1)/2 **entonces dev** x **sino dev** -1 **fsi** **ffun**
Pág. 125	En el bucle **mientras** en lugar de V (OR) debe poner ∧(AND)
Pág. 125	En el primer **if** del bucle **mientras** donde las dos abcisas son iguales, en lugar de **hb = a.altura** debe poner **hb = b.altura**
Pág. 126	En el cuerpo de los dos últimos bucles **mientras** del algoritmo después de ambas uniones debe sustituirse sa[ia] y sb[ib] por sa[ia++] y sb[ib++].
Pág. 157	El coste temporal, dado por el tamaño de la tabla, es el mismo. Debe ser: El coste espacial, dado por el tamaño de la tabla, es el mismo.
Pág. 157	En el algoritmo, dentro de la instrucción caso de: - C[i,j] = C[i-1,j]+1 hacer debe ser i > 0 y C[i,j] = C[i-1,j]+1 hacer - C[i,j] = C[i,j-1]+1 hacer debe ser j > 0 y C[i,j] = C[i,j-1]+1 hacer - C[i,j] = C[i-1,j-1]+1 hacer debe ser i> 0 y j > 0 y C[i,j] = C[i-1,j-1]+1 hacer - C[i,j] = C[i-1,j-1] hacer

	debe ser i> 0 y j > 0 y C[i,j] = C[i-1,j-1] hacer
Pág. 174	Subconjuntos de suma dada Donde dice enteros sin repeticiones (tanto positivos como negativos) -> enteros positivos sin repeticiones
Pág. 175	v[k] <- cierto **si** suma + datos[k] <= C **entonces** debe ser **si** suma + datos[k] <= C **entonces** v[k] <- cierto
Pág 177	**tipo** Vector = matriz[0..N] de entero debe ser **tipo** Vector = matriz[1..N] de entero
Pág 177	v[k+1] <- 1 **si** Completable(x, suma1, sumaTotal, k+1) **entonces** suma1 <- suma1 + x[k+1] DividirSociedad(x, suma1, suma2, sumaTotal, k+1, v) **fsi** v[k+1] <- 2 **si** Completable(x, suma2, sumaTotal, k+1) **entonces** suma2 <- suma2 + x[k+1] DividirSociedad(x, suma1, suma2, sumaTotal, k+1, v) **fsi** debe ser v[k+1] <- 1 **si** Completable(x, suma1, sumaTotal, k+1) **entonces** DividirSociedad(x, suma1+ x[k+1], suma2, sumaTotal, k+1, v) **fsi** v[k+1] <- 2 **si** Completable(x, suma2, sumaTotal, k+1) **entonces** DividirSociedad(x, suma1, suma2 + x[k+1], sumaTotal, k+1, v) **fsi**
Pág. 196	*El proceso continúa mientras la estimación optimista de la cima del montículo sea mayor que la cota de poda.*

	debe ser *El proceso continúa mientras la estimación optimista de la cima del montículo no sea menor que la cota de poda.*
Pág. 199	costeT <- costeT.valorT debe ser costeT <- hijo.costeT
Pág. 209	Distancia de edición Sobra la restricción con n <= m

Pág. 213	Distancia de edición
	
	estimacion <- costeT + costeInserción * (lonY-lonX)
	
	debe ser
	
	si lonX < lonY entonces
	  estimacion <- costeT + costeInserción * (lonY-lonX)
	sino
	  estimacion <- costeT + costeBorrado * (lonX-lonY)
	
	El comentario:
	
	Como explicábamos antes, la estimación optimista supone que todos los caracteres que faltan por comprobar entre las dos cadenas coinciden, y que sólo es necesario insertar los que faltan para llegar a la longitud de la cadena destino.
	
	debe ser
	
	Como explicábamos antes, la estimación optimista supone que todos los caracteres que faltan por comprobar entre las dos cadenas coinciden, y que sólo es necesario insertar los que faltan **o borrar los que sobran** para llegar a la longitud de la cadena destino.