Estructuras de Datos y Algoritmos

Doble Grado, Ingeniería Informática, de Computadores y del Software

Examen Primer Cuatrimestre, 11 de Febrero de 2016.

1. (3 puntos) Diseñar y verificar, o bien derivar, un algoritmo iterativo eficiente que cumpla la siguiente especificación:

$$\{0 \leq n \leq \mathit{longitud}(v) \wedge \forall k : 0 \leq k < n : v[k] \geq 0\}$$
$$\textbf{fun } \mathit{parImpar}(\textbf{int } v[\,], \textbf{int } n) \textbf{ return int } p$$
$$\{p = \#\ i, j : 0 \leq i < j < n : v[i]\,\%2 = 0 \wedge v[j]\,\%2 = 1\}$$

Indicar y justificar la complejidad del algoritmo.

2. (3,5 puntos) El ISBN de un libro consta actualmente de 13 dígitos $d_1\ d_2 \ldots d_{13}$. Al dígito d_{13} se le denomina dígito de control y se calcula a partir de los anteriores de la siguiente forma: primero se calcula la suma de multiplicar cada dígito de una posición par por 3 y cada dígito de una posición impar por 1. Los dígitos se numeran desde el 1 empezando por el más significativo:

$$d_1 * 1 + d_2 * 3 + d_3 * 1 + d_4 * 3 + d_5 * 1 + d_6 * 3 + d_7 * 1 + d_8 * 3 + d_9 * 1 + d_{10} * 3 + d_{11} * 1 + d_{12} * 3$$

El dígito d_{13} será aquel que sumado con el resultado de dicha suma de un múltiplo de 10. Por ejemplo, el dígito de control del número 336772900480 es 9 porque la suma anterior da como resultado 81 y $81 + 9 = 90$, que es múltiplo de 10.

Diseñar un algoritmo recursivo eficiente que ayude a calcular el dígito de control de un ISBN con cualquier número de dígitos. Igual que en el ejemplo, los dígitos anteriores vienen dados como un número entero. Por ejemplo, si el número dado es 2561, la suma será $2*1+5*3+6*1+1*3 = 26$ y por tanto el resultado sería 4. Indicar y justificar la complejidad del algoritmo.

3. (3,5 puntos) Consideremos una matriz, M, de números reales de dimensiones $N \times N$. Supongamos que esta matriz nos sirve para cartografiar cierto terreno que se ha dividido en forma de cuadrícula, de forma que $M[i][j]$ es la altura media de la casilla (i, j).

Se desea construir una carretera entre dos puntos dados. Para que una carretera pueda unir dos casillas adyacentes (no en diagonal) es necesario que la diferencia entre sus alturas medias no supere cierto valor, h_{max}, es decir:

$$\mid M[a][b] - M[c][d] \mid \leq h_{max}$$

donde las casillas (a, b) y (c, d) son adyacentes.

Dada una matriz M, una altura máxima h_{max} y dos casillas del terreno distintas entre sí, diseña un algoritmo de vuelta atrás que calcule la carretera de menor longitud (si es que existe) que resuelva el problema. Se considera que la longitud de la carretera es el número de casillas por las que pasa, incluyendo tanto el origen como el destino.

Por ejemplo, si la matriz M de alturas es la siguiente y $h_{max} = 2$:

$$\begin{pmatrix} 0 & 3 & 4 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$

la carretera de menor longitud para ir desde la casilla $(0, 1)$ hasta la $(0, 0)$ es de longitud 4 y pasa por las casillas $(0, 1)$, $(1, 1)$, $(1, 0)$, $(0, 0)$. Hay otras carreteras válidas pero no óptimas, como por ejemplo la que recorre las casillas $(0, 1)$, $(1, 1)$, $(2, 1)$, $(2, 0)$, $(1, 0)$, $(0, 0)$, que es de longitud 6. Una carretera no válida sería la que va directamente de $(0, 1)$ a $(0, 0)$ al contener una diferencia de alturas entre casillas adyacentes superior al máximo permitido.