Algorítmica y Complejidad – Grupo Tarde
22 Junio 2016 – Parte 2ª

Duración del examen: **1 hora 45 minutos** – Ponderación del examen: 60% de la nota

Ejercicio 1: 5 puntos
Ejercicio 2: 5 puntos

Fecha prevista de publicación de notas: 27 de Junio, por la tarde, en el sitio web habitual.
Revisión: 29 de Junio, el horario de revisión se publicará junto con las notas.

Ejercicio 1. Un cuadro latino de orden N es una matriz $N \times N$ donde las filas son permutaciones del conjunto $1..N$, y tal que cualquier entero k $(1 \leq k \leq N)$ aparece en cada fila y columna exactamente una vez. **Programar**, utilizando backtracking, un algoritmo que imprima todos los cuadros latinos de orden N.

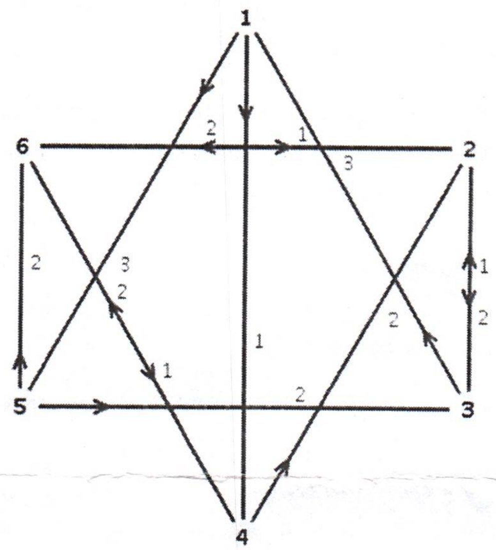
Ejercicio 2. Consideramos el grafo G de la figura.

a) Calcular la matriz $D(G) = (d_{ij})$ de distancias.
$d_{ij} =$ peso del camino mínimo que conecta i con j.

b) En el grafo G de la figura, denotamos por $C(i,j)$ el camino de peso mínimo que conecta el nodo i con el nodo j. Hallar $C(4,k)$ para $1 \leq k \leq 6$.

c) Sea X un grafo cualquiera con pesos no negativos y supongamos que la matriz $D(X)$ es simétrica. Probar que esta propiedad no implica necesariamente que X sea un grafo no dirigido.

d) Sea X un grafo simple cuya matriz de adyacencia es triangular. Demostrar que X no tiene ciclos.



Algorítmica y Complejidad – Grupo Tarde
22 Junio 2016 – Parte 1ª

Duración del examen: **1 hora 45 minutos** – Ponderación del examen: 40% de la nota

Ejercicio 1:	2	puntos	Ejercicio 4:	3,5	puntos
Ejercicio 2:	2	puntos			
Ejercicio 3:	2,5	puntos			

Fecha prevista de publicación de notas: 27 de Junio, por la tarde, en el sitio web habitual.
Revisión: 29 de Junio, el horario de revisión se publicará junto con las notas.

Ejercicio 1. Supongamos que $f(n) > 1$, $h(n) > 1$ y que $O\big(f(n)\big) = O(h(n))$ cuando $n \to \infty$. ¿Es cierto que $O(\log f(n)) = O(\log h(n))$? ¿Y el recíproco?

Ejercicio 2. Diremos que un entero $k \geq 1$ es triangular si $k = 1 + 2 + 3 + \cdots + n$ para algún $n \geq 1$. Implementar un algoritmo con complejidad $\Theta(\sqrt{k})$ que decida si un número entero $k \geq 1$ es triangular. Demostrar que el algoritmo satisface la complejidad requerida.

Ejercicio 3. Un punto de equilibrio de un array $A[1..n]$ de n componentes enteras es un índice $1 \leq p \leq n$ tal que la suma de las componentes de A situadas a la izquierda de p coincide con la suma de las que están situadas a la derecha, es decir,

$$\sum_{k=1}^{p-1} A[k] = \sum_{k=p+1}^{n} A[k].$$

Diseñar un algoritmo con complejidad $\Theta(n)$ que encuentre todos los puntos de equilibrio del array A.

Ejercicio 4. Sea A un array de n enteros positivos que, eventualmente, puede estar desordenado.

a) **Programar** un algoritmo divide y vencerás que calcule el máximo común divisor de los elementos de A.

b) Calcular la complejidad del algoritmo del apartado a). ¿Es mejor esta complejidad que la que se obtiene empleando la solución iterativa obvia, con independencia del contenido de A? Justificar la respuesta.