



## Examen Julio 2018, preguntas

Algoritmos (Universidade da Coruña)

**Examen de Algoritmos**  
**Julio de 2018**

<b>Apellidos:</b>	
<b>Nombre:</b>	<b>DNI:</b>

1. (2.5 puntos) Montículos:

- a) Presente el contenido de un **montículo de máximos** después de la llamada a `crear_monticulo` con un vector que contiene la siguiente secuencia de claves: 5, 1, 2, 3, 9, 6, 4, 7, 8.
- b) Desarrolle el pseudocódigo de la operación `crear_monticulo` **con complejidad  $O(n)$**  partiendo de la siguiente declaración para la implementación de montículos en base a vectores.

```

tipo Monticulo = registro
    Tam : 0..Max
    Vec : vector [1..Max] de Tipo_elem
fin registro

procedimiento Inicializar_Monticulo ( M )
    M.Tam := 0
fin procedimiento
  
```

Presente también el pseudocódigo de todos los procedimientos auxiliares necesarios.

- c) Indique la complejidad computacional de todas las operaciones **razonando y demostrando su respuesta sobre el pseudocódigo**.

2. (3 puntos) Ordenación:

- (a) Compare la *ordenación por fusión* con la *ordenación rápida* desde el punto de vista del diseño del algoritmo (técnica de diseño, fases de la técnica utilizada, estrategias para la mejora del tiempo de ejecución, ...) así como su complejidad (relaciones de recurrencia según cada caso, que se justificarán y se resolverán, teorema necesario para resolverlas, ...).
- (b) Muestre el estado del siguiente vector ( $n=14$ ) tras cada una de las fases de su ordenación mediante el *algoritmo de Shell* con *incrementos de Shell*:

10	4	9	7	8	5	13	3	0	6	1	14	2	11
----	---	---	---	---	---	----	---	---	---	---	----	---	----

3. (2 puntos) Dado el *grafo dirigido*  $G = (N, A)$ , con  $N = 1, 2, 3, \dots, 8$  y  $A$  determinado por la lista de aristas siguiente (se indica cada arista y su peso, o distancia):

1,2	1,3	2,7	3,2	3,4	3,5	3,7	4,5	4,8	5,6	5,8	7,5	7,6	8,6
1	2	1	6	2	7	2	3	4	5	2	1	2	1

Dibuje el grafo evitando el cruce de sus aristas, y a continuación dibuje:

- (a) Un árbol asociado a un recorrido en profundidad a partir del nodo 1.
- (b) Un árbol expandido mínimo del grafo no dirigido subyacente, indicando también su peso total.
- (c) Un árbol con caminos mínimos entre el nodo 1 y los demás, asociando a cada nodo la distancia mínima calculada desde el origen.
- (d) Una modificación gráfica del grafo en la que sus nodos se encuentren alineados y ordenados topológicamente.

4. (2,5 puntos) Se dispone de  $n$  objetos no fraccionables  $1, 2, 3, \dots, n$ , caracterizados por su peso  $w[i]$  y su valor  $v[i]$ . Una función *Mochila* permite encontrar, aplicando la técnica de *Programación Dinámica*, la carga de valor máximo ( $V$ ) para una mochila limitada por el peso máximo que puede contener ( $W$ ): para ello, calcula una tabla  $M[1..n, 0..W]$  con todos los resultados intermedios  $M[i, j]$  ( $i$  objetos, capacidad  $j$ ). La solución  $V = M[n, W]$  puede corresponder a una o varias configuraciones de la carga, constituidas por uno o varios conjuntos de objetos, sabiendo que no caben todos.

Se plantea el diseño de una función *Composición* que, a partir de la tabla  $M$  construida por la función *Mochila*, devuelva una configuración posible de la solución, especificando el conjunto de objetos que la componen:

- Proponga un ejemplo del problema, limitándolo a 3 o 4 objetos.
- Proponga un tipo de datos adecuado para la salida de la función *Composición*, que ilustrará con el problema anterior.
- Proponga un pseudocódigo de la función *Composición*.
- Determine su complejidad.