	Original			
				Nacional y U.E.

INDICACIÓN DE LAS SOLUCIONES:

Cuestión 1 (1 punto). Describe el grafo asociado al espacio de soluciones del problema del *n-puzzle* (práctica de este curso).

Solución:

- Cada nodo se bifurca como máximo en los cuatro movimientos posibles del hueco (arriba, abajo, izquierda, y derecha), siempre que sea posible.
- La profundidad del árbol está limitada por el número de distintos tableros posibles ((nxn)!, siendo n el tamaño del lado del tablero), ya que no se deben repetir.

Cuestión 2 (2,5 puntos). Supongamos que un polinomio se representa por un vector *v[0..n]* de longitud *n*+1 donde el valor *v[i]* es el coeficiente de grado *i*. Describir claramente los cálculos y la descomposición algorítmica necesaria para plantear un algoritmo, de orden mejor que cuadrático, que multiplique dos polinomios P(x) y Q(x) mediante divide y vencerás. NOTA: La solución típica trivial de orden cuadrático puntúa 0 puntos.

Solución:

Es relativamente simple dar con la solución de orden cuadrático. En este caso, se descompone en mitades $P(x) = Ax^{n/2}+B$ y $Q(x) = Cx^{n/2}+D$ con A,B,C,D polinomios de grado n/2 sacando factor común $x^{n/2}$ como se detalla en las expresiones. De esta forma se ve claramente que $P\cdot Q$ es $(Ax^{n/2}+B)\cdot(Cx^{n/2}+D) = AC\,x^{n} + (AD+BC)x^{n/2} + BD$ lo que la solución conlleva 4 multiplicaciones de grado n/2. El coste sería en este caso cuadrático. Sin embargo hay una manera de organizar las operaciones mediante la cual, no es necesario calcular AD+CD mediante 2 productos, sino sólo con uno, aprovechando que ya tenemos realizados los productos BD y AC. En este último caso basta con observar que $(A+B)\cdot(C+D) = AC+BC+AD+BD$ y que $BC+AD = (A+B)\cdot(C+D) - AC - BD$ con lo que es posible realizar el cálculo con 3 productos en lugar de 4, ya que el coste de las sumas, si consideramos las multiplicaciones como lineales, sería constante. De manera que $(A+B)\cdot(C+D)$ sería uno de los productos, y AC y BD los otros dos.

Este problema es similar al problema de la multiplicación de números enteros muy grandes del texto base, capítulo 7.

Cuestión 3 (2,5 puntos). Con respecto al algoritmo de ordenación por fusión (Mergesort).

a) Escribe el algoritmo.

b) Dibuja la secuencia de llamadas del algoritmo y la evolución del siguiente vector al ordenarlo. Para ello ten en cuenta que hay que considerar que el problema es suficientemente pequeño cuando el subarray es de tamaño 2.

(2,0,2,1,9,6,2,3,5,8)

Solución:

(a) El algoritmo se puede encontrar en Brassard & Bratley, pag. 258

(b)

```
OrdFusion(2,0,2,1,9,6,2,3,5,8)
        OrdFusion(2,0,2,1,9)
                OrdFusion(2,0)
                        Insertar(2,0) -> (0,2)
                OrdFusion(2,1,9)
                        OrdFusion(2)
                                Insertar(2) -> (2)
                        OrdFusion(1,9)
                                Insertar(1,9) -> (1,9)
                        Fusionar((2),(1,9)) -> (1,2,9)
                Fusionar((0,2),(1,2,9)) -> (0,1,2,2,9)
        OrdFusion(6,2,3,5,8)
                OrdFusion(6,2)
                        Insertar(6,2) -> (2,6)
                OrdFusion(3,5,8)
                        OrdFusion(3)
                                Insertar(3) -> (3)
                        OrdFusion(5,8)
                                Insertar(5,8) -> (5,8)
                        Fusionar((3),(5,8)) -> (3,5,8)
                Fusionar((2,6),(3,5,8)) -> (2,3,5,6,8)
        Fusionar((0,1,2,2,9), (2,3,5,6,8)) -> (0,1,2,2,2,3,5,6,8,9)
```

Problema (4 puntos). Tenemos n objetos de volúmenes $v_1 \ldots v_n$, y un número ilimitado de recipientes iguales con capacidad R (con $v_i \leq R$, $\quad i$). Los objetos se deben meter en los recipientes sin partirlos, y sin superar su capacidad máxima. Se busca el mínimo número de recipientes necesarios para colocar todos los objetos.

La resolución de este problema debe incluir, por este orden:

1. Elección razonada del esquema algorítmico más apropiado para resolver el problema. Escriba dicho esquema general.(0,5 puntos).
2. Descripción de las estructuras de datos necesarias (0,5 puntos).
3. Algoritmo completo a partir del refinamiento del esquema general (2 puntos).
4. Estudio del coste del algoritmo desarrollado (1 punto).

Solución:

Resuelto en Estructuras de datos y métodos algorítmicos. N.Martí, Y. Ortega, J.A Verdejo.
Pearson education, 2004. Ejercicio 15.9.

Indicaciones:

1. Ramificación y poda.

Se trata de un problema de optimización, para el que no existe una función de selección que permita ir seleccionando a cada paso el objeto que de lugar a la construcción parcial de la solución óptima. Por tanto no es posible aplicar un algoritmo voraz. Tampoco existe una forma de dividir el problema en subproblemas que se puedan resolver independientemente, por lo que tampoco es posible un esquema divide y vencerás.

2.
vector de objetos:
Podemos representar el reparto de objetos entre recipientes mediante un vector en el
que cada posición indique a qué recipiente se ha asignado el objeto correspondiente.

objetos = vector[1..n] of entero

La solución es la cantidad entera S de recipientes empleados.

Montículo de mínimos en el que cada componente almacene una solución parcial (nodo)
con su cota correspondiente.

nodo = tupla
  asignaciones: vector[1..N] de enteros
  etapa: cardinal // objeto por el que se va procesando
  num-recip: cardinal // num de recipientes
  capacidad[1..N] de real // capacidades disponibles de cada envase utilizado

3. Para el desarrollo se utilizan las siguientes estimaciones

cota inferior:
  el número de envases ya utilizados en la solución parcial

cota superior
Se hace una estimación metiendo cada objeto de los pendiente en el primer
recipiente en el que quepa. Si no cabe en ninguno se añade un nuevo envase a la solución parcial.

4. El número de recipientes está limitado a n, es decir al número de objetos.
Una estimación del coste es el tamaño del árbol, que en el peor caso crece como
O(n!), ya que cada nodo del nivel k puede expandirse con los n-k objetos que quedan
por asignar a recipientes.