Parcial 2 – 13 de abril de 2015

Apellidos, nombre ______________________________________ NIF: ________________

Pregunta 1 (4 puntos)

Los mundiales de natación de Barcelona 2013 han sido un gran paso para la natación española con 12 medallas en total. Una de las pruebas que se ha decidido mejorar para el próximo mundial es el relevo 4x100 estilos femeninos, en el que participan 4 nadadoras cada una en un estilo de natación diferente: espalda (E), braza (B), mariposa (M) y crol (C). El seleccionador dispone de cuatro nadadoras y de sus tiempos para cada una de las pruebas. Se quiere diseñar el algoritmo que mediante la técnica de backtracking permita decidir al seleccionador que nadadora participará en cada estilo de tal forma que optimicen los tiempos del conjunto.

a) (3 puntos) Completar el código Java para dar solución a este problema (escribirlo en los recuadros preparados a tal efecto).

```java
static int nEstilos;
static int[][]tiemposNadadoras;  // tiempos de las nadadoras
static boolean [] marca;

static int[]seleccion;
static int tiempo;

static int[]seleccionMejor;// mejor selección
static int tiempoMejor;// tiempo de la mejor


static void backtracking (________________________)
{
        if (________________________)
        {
                if (________________________)
                {
                        ________________________
                        ________________________
                        ________________________
                }
        }
        else
        {
                for (________________________)
                        if (________________________)
                        {
                                ________________________
                                ________________________
                                marca[est]=true;
                                backtracking (______________);
                                ________________________
                                ________________________
                        }
        }
}
```

b) (1 punto) Razonar si esta técnica es eficiente para realizar el cálculo y qué otras técnicas vistas en clase podrían dar solución a este problema indicando las ventajas y los inconvenientes.

Pregunta 2 (3 puntos)

El algoritmo de Floyd consiste en encontrar el camino simple más corto entre todos los pares de nodos de un grafo. Dado un grafo G, lo que queremos es hallar el camino más corto para ir desde un nodo i hasta otro nodo j de forma directa o a través de otro nodo k.

Para resolver este problema tenemos disponible la matriz de pesos del grafo *G(i,j)*, es decir, una matriz que recoge el coste de ir de un nodo *i* a otro *j* del grafo, sin pasar por nodos intermedios. Cuando no hay camino directo entre dos nodos, la celda correspondiente tendrá valor infinito. Se aplica la siguiente función *D(i,j)= min(D(i,j), D(i,k)+D(k,j))* pudiendo ser *k* cualquier otro nodo del grafo.

a) Razonar por qué decimos que Floyd se encuadra dentro de las técnicas de programación dinámica.
b) Explicar que valores se pueden rellenar de forma directa en la tabla.
c) Marcar sobre el array del apartado b) las casillas de las que depende (2,3) para calcular su valor.
d) (1 punto) Rellenar la matriz D y la matriz de P de caminos que permite recuperar la secuencia de nodos para el camino más corto entre dos nodos con los siguientes datos de entrada:

Orig/Dest	1	2	3	4
1	∞	3	4	2
2	∞	∞	∞	3
3	1	4	∞	∞
4	-1	∞	6	∞

e) Devolver el valor y la secuencia de nodos obtenida en el apartado anterior, cuando se va desde el nodo 2 al nodo 3.

Pregunta 3 (3 puntos)

Tenemos una colección de **n** ficheros que nos resultan de mucho valor, aunque lamentablemente necesitaremos borrar alguno de ellos porque nos hemos quedado sin espacio en nuestro disco duro de 600 MB.

Por ese motivo, tenemos que ingeniárnoslas para tratar que los ficheros con los que finalmente nos quedamos en el disco duro tengan el máximo valor posible para nosotros (los cuantificaremos en base a lo importantes que sean para nosotros). No podemos dividir los ficheros en partes más pequeñas porque dejarían de funcionar en nuestro ordenador, por lo que o nos quedamos con un fichero o lo descartamos.

a) Indica tres heurísticos que podíamos utilizar en un algoritmo voraz para calcular con qué ficheros nos quedamos.
b) Indica si dichos algoritmos son óptimos o no y demuéstralo con un contraejemplo.