

Estructura de Datos y de la Información

2º – ITIS y 3º Teleco + ITIS

Práctica Obligatoria

Curso 2009/10

El alumno debe responder de manera razonada a las diferentes cuestiones planteadas. Se entregará una **memoria en papel** con el trabajo desarrollado como se pide en el enunciado así como los códigos desarrollados en soporte óptico o magnético para su corrección. El código fuente entregado deberá ser compatible con la versión de Turbo Pascal 7.0 o con EclipseGavab.

Fechas de entrega: coincidirá con la del examen de la convocatoria correspondiente:

- Convocatoria de enero: 18 de enero de 2010
- Convocatoria ordinaria: 19 de mayo de 2010
- Convocatoria de junio: 24 de junio de 2010

1. Enunciado. Resolver un Survo Puzzle

Survo puzzle es un tipo de puzzle lógico desarrollado y presentado por Seppo Mustonen en 2006, que tiene cierta similitud con el popular sudoku. En este puzzle la tarea consiste en rellenar una matriz de $n \times m$ con los números enteros $1, 2, \dots, n \cdot m$ de modo que cada uno de ellos aparezca solamente una vez y que la suma por filas y por columnas sea igual a los números enteros dados en la última fila y la última columna de la tabla (ver figura 1). Al igual que en el sudoku, el problema puede tener algunos números iniciales como dato (como en el ejemplo de la figura 1.a), que ayudan a la resolución del problema y/o restringen las soluciones posibles a una sola.

	A	B	C	D	
1		6			30
2	8				18
3			3		30
	27	16	10	25	

	A	B	C	D	
1	12	6	2	10	30
2	8	1	5	4	18
3	7	9	3	11	30
	27	16	10	25	

Figura 1: Ejemplo de Survo Puzzle de 3x4: (izqda) situación inicial y (dcha) una vez resuelto. Observar que en la solución final no se repite ningún número y que la suma de los elementos por filas y columnas coincide con el entero sobre fondo gris.

2. Estudio del problema y diseño algorítmico.

Esta tarea pretende que el alumno estudie detenidamente el problema propuesto e identifique su solución con alguno de los esquemas algorítmicos propuestos en la asignatura. El alumno deberá explicar en qué estructuras de datos se va a basar, así como adaptar el esquema seleccionado para solucionar el problema propuesto.

Para el desarrollo correcto de la misma el alumno debe haber estudiado los esquemas algorítmicos vuelta atrás y ramificación y poda (Capítulo 9 del libro “Fundamentos de Algoritmia” y Capítulos 6 y 7 del libro “Técnicas de Diseño de Algoritmos”).

Detección del esquema algorítmico.

El alumno deberá responder razonadamente a las siguientes cuestiones:

1. ¿Por qué el esquema algorítmico más adecuado para solucionar el problema es un esquema basado en exploración de grafos? Razonar por qué otras aproximaciones no serían válidas.
2. ¿Qué tipo de estrategia de exploración de grafos (vuelta atrás o ramificación y poda) considera más adecuada para resolver su problema concreto? Razone su respuesta y explique las diferencias entre ambas.
3. Presentar y explicar el esquema algorítmico de exploración de grafos finalmente seleccionado, describiendo la funcionalidad de los distintos elementos del mismo (funciones auxiliares).

Particularización del algoritmo sobre el problema concreto.

El alumno deberá responder razonadamente a las siguientes cuestiones:

1. ¿Cuáles son las estructuras de datos más adecuadas para resolver el problema? Exponga y explique las estructuras de datos principales para resolver el problema, así como aquellas estructuras de datos auxiliares necesarias para almacenar la información de cálculos intermedios del algoritmo.
2. Adapte el esquema algorítmico seleccionado a su problema concreto. Especifique, diseñe y explique todas aquellas funciones auxiliares necesarias para la resolución del mismo.
3. Presente el diseño del algoritmo completo.
4. Realice un estudio teórico de la complejidad del algoritmo diseñado. Para ello deberá calcular y presentar separadamente el coste de todas y cada una de las funciones auxiliares implementadas en su algoritmo y, finalmente, calcular y presentar el coste total del algoritmo completo.

3. Implementación y prueba.

Esta tarea pretende que el alumno implemente, basándose en el diseño realizado en la tarea anterior y en las estructuras de datos propuestas, una función que resuelva el problema propuesto. Siendo necesario:

- a) Un módulo con el algoritmo diseñado, así como con todas las funciones auxiliares identificadas.
- b) Un módulo MAIN que, partiendo de un conjunto de datos de entrada proporcionados a éste desde un fichero de texto, haga las correspondientes llamadas al algoritmo para resolver el problema, mostrando la solución por pantalla.
- c) Todas las estructuras de datos empleadas en la resolución de la práctica serán implementadas y encapsuladas en sus propios módulos. Podrán añadirse tantos módulos accesorios como el alumno considere conveniente.

Implementación del algoritmo

El alumno deberá implementar el algoritmo principal de este primer bloque de trabajo respetando las siguientes restricciones:

1. Cree un módulo, de nombre SURVO, que contenga un subprograma, de nombre RESUELVE, que implemente el algoritmo diseñado y que devuelva la solución al problema.
2. Incluya en el módulo SURVO todos aquellos subprogramas auxiliares necesarios para el correcto funcionamiento del subprograma RESUELVE.

Implementación de un programa principal

El alumno deberá implementar un módulo principal MAIN que:

- a) Tomará como entrada de datos un fichero texto plano (ASCII), con nombre ENTRADA.TXT. Este fichero representará a una matriz inicial del juego de la siguiente forma:
 - En la primera fila, se podrán leer el número de filas (n) y de columnas (m) del problema separados por un espacio.
 - En las siguientes n filas del fichero aparecerán las sumas de las filas de la 1 a la n .
 - A continuación, en las siguientes m filas del fichero aparecerán las sumas de las columnas de la 1 a la m .
 - A continuación, los números iniciales en este formato (separados por un espacio):
número_de_fila número_de_columna número_inicial
- b) Mapeará la entrada leída a partir del fichero de datos sobre las estructuras de datos diseñadas y;
- c) Hará las correspondientes llamadas a la función RESUELVE, presentando el resultado obtenido en la pantalla.

ENTRADA.TXT

```
1 3 4
2 30
3 18
4 30
5 29
6 16
7 10
8 25
9 2 1 8
10 1 2 6
11 3 3 3
```

Figura 2: Ejemplo de fichero “ENTRADA.TXT” (en el recuadro) para el Survo Puzzle de la figura 1. En la fila 1 aparece el número de filas (3) y de columnas (4). En las siguientes tres filas (de la 2 a la 4) aparecen las sumas por filas (30 18 30). En las siguientes cuatro filas (de la 5 a la 8) aparecen las sumas por columnas (29 16 10 25). Las últimas tres filas (de la 9 a la 11) indican la posición y valor de los números iniciales: La fila 9 del fichero indica que en la fila 2 y columna 1 hay un 8, la fila 10 del fichero indica que en la fila 1 y columna 2 hay un 6 y, finalmente, la fila 11 del fichero indica que en la fila 3 y columna 3 hay un 3.

En la figura 2 se muestra el formato para ENTRADA.TXT correspondiente al ejemplo planteado en la figura 1. El programa principal MAIN deberá procesar esta información contenida en el fichero de entrada, mostrar dicha información por pantalla, indicando que se trata del survo puzzle a resolver, y mostrar el resultado devuelto por el subprograma RESUELVE.