# Descripción del Problema

Se pretende obtener un programa que permita al usuario calcular de forma automática la distribución más conveniente de la carga que se va a realizar sobre un buque carguero.

El buque puede tener un numero variable de compartimentos en su interior. Para su navegación sea lo más estable posible, se deberán llenar cada uno de los compartimentos con el mismo peso (o lo más cercano al mismo), de forma que entre ellos se mantenga el mayor equilibrio posible.

Se puede considerar que los compartimentos tendrán una capacidad igual (determinada por la capacidad total del barco). La estructura de la carga dentro de los compartimentos se considerará cuadrada: si, por ejemplo, se determina que se pretenden almacenar en cada uno de ellos 16 contenedores; se almacenan en un espacio de 4 filas y 4 columnas. Si se determina que se quieren guardar en cada contenedor son 6 contenedores, la estructura disponible será de 3 filas y 3 columnas, dejando la última fila vacía. Siempre se preferirá tener una base más amplia que el apilar a más alturas. Es decir, en caso de que la división no sea exacta, el compartimento será siempre más ancho que alto en una unidad. Por ejemplo: 17 contenedores  $\rightarrow$  5 base x 4 alto (más una fila vacía).

En cuanto a distribución de la carga se deben tener en cuenta dos características: el peso de cada contenedor y el puerto, dentro de la ruta del carguero, en el que se tiene previsto descargar.

De esta forma, se **prohíbe** que un contenedor de un determinado peso se cargue encima de otro de menor peso: debe situarse siempre sobre otro de un peso igual o mayor. Se trata de una restricción estricta, pues esa situación puede dañar el contenido de la carga.

Por otro lado, se **prefiere** que los contenedores que van a ser descargados en puertos anteriores estén almacenados por encima de aquellos otros que van a ser descargados en puertos situados detrás en la ruta. De esa forma, al alcanzar un determinado puerto, se evita tener que mover un contenedor que no se va a descargar para poder acceder a otro que sí.

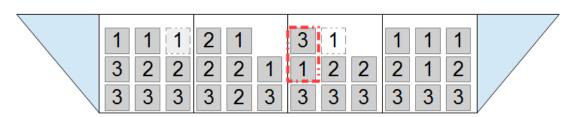


Ilustración 1. Organización de un problema de carguero en el que se representa el puerto de destino de cada contenedor.

En el ejemplo de la Ilustración 1, se marca en rojo un contenedor que se debe desembarcar en el 3º puerto que se visita colocado sobre otro que deberá descargarse en el 1º puerto. Esto obligará a mover el primer contenedor (arriba) para sacar el segundo (abajo); aun cuando deberá permanecer en el carguero y, por lo tanto, volverse a colocar.

### Formato de Ficheros

### Fichero de Entrada

Se espera que el programa pueda adaptarse a diferentes configuraciones del problema a resolver sin necesidad de modificaciones del código.

Para poder introducir al programa las diferentes configuraciones, los datos de cada configuración del problema se cargarán desde ficheros de texto plano. Estarán formateados siguiendo el formato de CSV. Es decir, se trata de filas de datos; separando cada columna por comas.

El fichero tendrá el siguiente formato:

#### Sección 1.

La primera línea del fichero contendrá los datos del navío sobre el que se trabaja. El primer número será el número de compartimentos disponibles en el carguero y el segundo el número de contenedores que se tendrán que cargar.

### Sección 2.

El resto de líneas del fichero, corresponderá cada una a cada contenedor a almacenar. Aparecerán, siempre en orden los siguientes datos del contenedor:

- Identificador del Contenedor: numero entero en el rango de 0 al número de contenedores a almacenar -1. Los contenedores aparecen en el fichero de forma secuencial.
- Peso (en toneladas): será un número entero positivo indicando su peso.
- Puerto de destino: entero indicando el orden de visita en el trayecto. El primero en ser visitado será el marcado con un 0, el segundo con un 1, etc.
- Mercancía peligrosa: booleano que indica si el contenedor almacena mercancías peligrosas. Se empleará este dato en sucesivas prácticas.

### Fichero de salida

Una vez realizados todos los cálculos, se solicita que el programa devuelva la solución que se considera como la más correcta alcanzada por el algoritmo.

Para ello, se pide que devuelva esta salida en otro fichero CSV, con los resultados en el siguiente formato:

- Sección 1. Contendrá una única línea en la que aparecerán los datos del número de compartimentos y número de contenedores a almacenar.
- Sección 2. Contendrá la distribución de contenedores en cada compartimento. Se incluirá una línea por compartimento. Se enumerarán los contenedores almacenados en él indicando su número de identificador (el que se les asigna en los datos de entrada). Se empezarán a enumerar los contenidos por la fila de almacenamiento más baja y de izquierda a derecha. En las posiciones en las que considere que no se incluirá ningún contenedor, se incluirá el identificador -1.

# **Ejemplos**

### Fichero de entrada

Dado un fichero de entrada con el siguiente contenido:

5,8 0,218,4,1 1,787,3,1 2,321,4,0 3,49,1,0 4,486,1,0 5,621,4,0 6,526,6,1 7,772,5,0

### Se interpretará como:

Línea 1: Se trata de un carguero con 5 compartimentos, en el que hay que almacenar 8 contenedores.

Línea 2: El contenedor 0: pesa 218 toneladas, se descarga en el 5º puerto a visitar y contiene mercancías peligrosas.

Línea 3: El contenedor 1: pesa 787 toneladas, se descarga en el 4º puerto a visitar y contiene mercancías peligrosas.

Línea 4: El contenedor 2: pesa 321 toneladas, se descarga en el 5º puerto a visitar y no contiene mercancías peligrosas.

Línea 5: El contenedor 3: pesa 49 toneladas, se descarga en el 2º puerto a visitar y no contiene mercancías peligrosas.

... etc.

### Fichero de salida

Dado un fichero de entrada con el siguiente contenido:

5,30 6,5,14,28,9,8,-1,18,-1 0,10,4,26,20,23,-1,-1,-1 16,12,13,2,21,29,-1,-1,-1 1,17,7,22,24,-1,-1,-1,-1 25,11,3,19,15,27,-1,-1,-1

Se interpretará como la siguiente configuración:



# UNIVERSIDAD DE BURGOS

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

# **Grado en Ingeniería Informática** Computación Neuronal y Evolutiva

<b>Compartimento</b>			Compartimento			Compartimento			Compartimento			Compartimento		
	<u>1</u>			<u>2</u>			<u>3</u>			<u>4</u>			<u>5</u>	
-	18	-	-	-	-	-	-	-	-	-	-	-	-	-
28	9	8	26	20	23	2	21	29	22	24	-	19	15	27
6	5	14	0	10	4	16	12	13	1	17	7	25	11	3