

# TreeKEA

Enunciado de la práctica de PRO2  
Primavera 2018

11 de mayo de 2018

## 1. Introducción

Una cadena de tiendas de muebles desea cambiar sus métodos de almacenaje probando nuevas distribuciones de sus almacenes. Nos ha encargado que implementemos una simulación de un nuevo tipo de almacén para poder estudiar si adoptan este nuevo diseño en futuras tiendas.

Esta empresa trabaja con un gran número de productos diferentes. Cada producto tiene un identificador que consta de una serie de letras y números. Las letras pueden ser tanto mayúsculas como minúsculas. El identificador comienza siempre por una letra. Se diferencian las mayúsculas de las minúsculas.

El nuevo tipo de almacén constará de  $n$  salas diferentes identificadas por los enteros  $1, \dots, n$ . Solo se puede entrar al almacén por una única sala, cuyo identificador puede ser cualquiera. Desde cualquier sala se puede avanzar directamente o bien a dos salas, una por el lado izquierdo y otra por el lado derecho, o bien a ninguna sala. Evidentemente el almacén es finito. Desde cualquier sala, menos la inicial, se puede retroceder directamente hasta una sala.

Cada sala tendrá una estantería con  $f$  filas y  $c$  columnas. Las dimensiones de las estanterías no son necesariamente iguales en todas las salas. Las dimensiones de una estantería se pueden cambiar en cualquier momento bajo ciertas condiciones.

Cualquier producto cabe en cualquier posición de la estantería.

Diremos que una posición  $p_1$  de la estantería *va antes* que una posición  $p_2$  si la fila de  $p_1$  es más baja que la de  $p_2$  o en el caso en que las filas sean iguales la columna de  $p_1$  está a la izquierda de la columna de  $p_2$ , es decir, la posición  $i, j$  va antes que la posición  $k, l$  si  $i > k$  o bien si  $i = k$  y  $j < l$ .

Por ejemplo en la siguiente estantería

NULL	AAAA	BBBB
CCCC	DDDD	EEEE
FFFF	GGGG	HHHH
VVVV	YYYY	ZZZZ

la posición 1,1 está ocupada por NULL, la posición 4,3 por ZZZZ y la posición 2,2 por DDDD.

Para colocar un producto en una estantería se usará el hueco (posición sin producto) que vaya antes. Para quitar un producto determinado, siempre se quitará la aparición del producto que vaya antes. Naturalmente a base de poner y quitar productos en una estantería pueden aparecer huecos que vayan antes que productos que ya estén colocados.

A veces se reestructurarán las estanterías siguiendo dos criterios. El primero de ellos es la *compactación*, en la que desplazaremos los productos hacia la izquierda y hacia abajo de manera que no quede ningún hueco entre dos productos, ni antes de ningún producto. No se ha de cambiar el orden relativo entre los productos. Si un producto  $p_1$  va antes que  $p_2$  antes de la compactación, sigue yendo antes después de la compactación. La otra forma de reestructurar una estantería es la *reorganización* en la que ordenaremos los productos por alfabéticamente, sin dejar huecos entre ellos ni antes de ningún producto. Tanto en la compactación como en la reorganización, si la estantería no está llena, quedarán huecos después de los productos.

También se pueden cambiar las dimensiones de una estantería en cualquier momento. Para poder hacerlo es necesario que con las nuevas dimensiones quepan todos los productos. Si caben los productos se puede visualizar el resultado final como si pusiéramos en la sala una estantería con las nuevas dimensiones y moviésemos los productos de la estantería inicial a la nueva manteniendo el orden relativo de los productos, sin dejar huecos antes del primer producto ni entre productos. Luego nos llevamos la estantería inicial, que se ha vaciado.

Se pueden añadir una cierta cantidad de productos en una determinada sala, o quitarlos. Se pueden distribuir productos por todo el almacén. Se podrían implementar diferentes políticas de distribución, pero de momento, para hacer la simulación solo implementaremos una que consiste en, empezando por la sala inicial, poner todos los productos que quepan en la sala en que estemos e intentar distribuir el resto en las siguientes salas del almacén. Si el número de las unidades que faltan por distribuir es par se distribuye

la mitad a partir de la siguiente sala por la izquierda y la otra mitad a partir de la siguiente sala por la derecha, si es impar, la unidad “sobrante” se distribuye por la izquierda.

Queremos poder hacer varios tipos de consultas. Se ha de poder consultar que producto hay en una determinada posición de la estantería de una sala. También queremos poder escribir el contenido de la estantería de una sala y decir cuantas unidades de cada producto hay en la sala y cuantas posiciones están ocupadas en la estantería. También queremos saber cuantos unidades de cada producto hay en total en el almacén o cuantas unidades de un determinado producto hay en total en el almacén.

Para trabajar con un producto (quitarlo, ponerlo, etc) es necesario haber proporcionado su identificador al sistema. Se puede borrar un producto del sistema siempre y cuando no quede ninguna unidad en el almacén. Más adelante se puede volver a introducir en el sistema si así se desea.

## 2. Detalle de las operaciones

En este apartado listamos las operaciones de nuestro sistema. Para ver los detalles concretos del formato conviene consultar el juego de pruebas público. Primero se leerá el número de salas del almacén y luego la estructura del almacén en preorden. Luego por orden de identificador de sala se leerán las dimensiones (filas, columnas) de la estantería de la sala.

Los identificadores de sala serán siempre válidos, es decir, su valor estará entre 1 y  $n$  incluidos. Las cantidades de producto siempre serán positivas. Los valores de las filas y las columnas siempre serán mayores que cero y cuando se apliquen a una posición de una estantería, dicha posición existirá en ese momento.

He aquí las diferentes operaciones:

- **poner\_prod**: Se introduce un identificador de producto. Si el producto ya existía se produce un error, sino el producto se da de alta en el sistema con cero unidades.
- **quitar\_prod**: Se introduce un identificador de producto. Si el producto no existe, o ya existe y quedan unidades se produce un error, sino el producto se da de baja en el sistema.
- **poner\_items**: Se introduce un identificador de sala, un identificador de producto y una cantidad. Si el producto no existe se produce un error. Se colocan tantas unidades como quepan en la sala y se devuelve

un entero que indique cuantas unidades no han cabido. Se colocarán las unidades lo antes posible, es decir, primero se rellenan los huecos a partir del hueco que vaya antes. Las unidades que no quepan no se guardan en el almacén.

- **quitar\_items**: Se introduce un identificador de sala, un identificador de producto y una cantidad. Si el producto no existe se produce un error. Se quitan tantas unidades como se pueda y se devuelve un entero que indique cuantas unidades no se han podido quitar porque no había suficientes en la sala. Se comenzará quitando las unidades que vayan antes.
- **distribuir**: Se introduce un identificador de producto y una cantidad. Si el producto no existe se produce un error. Se distribuye el producto según la política explicada previamente. Se devuelve un entero que indica cuantas unidades no se han podido distribuir. Las unidades que no se distribuyan no se guardan en el almacén.
- **compactar**: Se introduce un identificador de sala y se compacta su estantería.
- **reorganizar**: Se introduce un identificador de sala y se reorganiza su estantería.
- **redimensionar**: Se introduce un identificador de sala y el nuevo número de filas y el nuevo número de columnas. Si los productos que hay en la sala no caben en las nuevas dimensiones se produce un error. Se redimensiona la estantería de la sala.
- **inventario**: No tiene datos. Para cada producto en el sistema se escribe su identificador y la cantidad total en el almacén, ordenado por identificador de producto.
- **escribir**: Se introduce un identificador de sala. Se escribe el contenido de la estantería de arriba a abajo y de izquierda a derecha. En los huecos escribiremos NULL y por tanto no podrá ser un identificador válido de producto. También escribiremos cuantas unidades hay en total y por orden de identificador de producto existente en la estantería, escribiremos el identificador de producto y su cantidad.
- **consultar\_pos**: Se introduce un identificador de sala, la fila y la columna. Se indica que producto hay en la posición correspondiente de la estantería de dicha sala. Si no hay producto se escribe NULL.

- **consultar\_prod**: Se introduce un identificador de producto. Si no existe, se produce un error. Si existe se escribe cuantas unidades hay en total en el almacén.
- **fin**: Sin datos. Se acaba la ejecución de la simulación.

Para facilitar la legibilidad se escribirán todas las instrucciones a partir de la primera columna y todos sus parámetros separados por un espacio. Los resultados se escribirán a partir de la columna 3. Para más detalles se puede consultar el juego de pruebas público.