

Estructuras de Datos
Curso 2016-2017. Convocatoria de Febrero
Grado en Ingeniería Informática.
Doble Grado en Ingeniería Informática y Matemáticas

1. (1 punto) Es posible mantener 2 pilas en un vector (denominaremos a la estructura piladoble) si una crece desde la posición 0 (pila 0) del vector y la otra decrece desde la última posición (pila 1). Dar una especificación (sintaxis y semántica) de ese nuevo TDA. Implementar una función **void pop(int indicepila)** que aplicada a una piladoble **p** (**p.pop(0)** ó **p.pop(1)**), borre un elemento de la pila referenciada por indicepila (0 ó 1)
2. (1 punto) Se define una **matriz bilineal simétrica** como una **matriz nxn** en la que todos los elementos significativos (distintos de un valor por defecto) están situados en las 2 diagonales principales y tal que al recorrer ambas diagonales en orden creciente de filas, presentan los mismos elementos. Un ejemplo de este tipo de matrices es la matriz del ejemplo. Es una matriz **6x6** de valores enteros y con el **0** como valor por defecto.

2	0	0	0	0	2
0	4	0	0	4	0
0	0	7	7	0	0
0	0	1	1	0	0
0	9	0	0	9	0
6	0	0	0	0	6

Implementar un iterador que itere sobre los elementos significativos de la matriz. Han de implementarse (aparte de las de la clase iteradora) las funciones **begin()** y **end()**.

3. (1 punto) Tenemos un contenedor de pares de elementos, {clave, vector<int>} definida como:

```
template <typename T>
class contenedor{
private:
    unordered_map<T, vector<int> > datos;
    .....
    .....
}
```

Implementa una función que inserte un nuevo elemento en el contenedor y otra que cambie un elemento concreto (en una posición p) del vector

4. (1 punto) Sea A un **APOmin** (APO con el orden parcial \leq). Diseñar un procedimiento **borramax** (A) que elimine el mayor elemento almacenado en A de forma que la estructura final siga siendo un APOmin.
5. (1 punto) Dado un **arbol binario de búsqueda A**, implementar una función, que, nos permita conocer el siguiente nodo a un **nodo** dado **n** en inorden, es decir, implementar el **operator++** del iterador inorden del ABB
6. (1 punto) Se define la trayectoria de una hoja en un árbol binario como la suma del contenido de todos los nodos desde la raíz hasta la hoja multiplicada por el nivel en cada nodo se encuentra. Implementar (usando el **TDA bintree**) un procedimiento que, dado un árbol binario devuelva la hoja con mayor trayectoria

<pre> 7 / \ 1 3 / \ / \ 4 2 1 </pre>	<p>Tra(4)=4*3+1*2+7*1=21 (**)</p> <p>Tra(2)=2*3+3*2+7*1=19</p> <p>Tra(1)=1*3+3*2+7*1=16</p>
---	---

Tiempo: 3 horas