



1. (0.75 puntos) **(1)** La definición **set <stack <int>> :: iterator p;**

(1-a) Es incorrecta porque las pilas no tienen iteradores

(1-b) Es incorrecta porque no se puede iterar sobre un conjunto de pilas

(1-c) Es incorrecta porque no puede definirse un conjunto de pilas

(1-d) Es correcta

Razona la respuesta

- (2)** Dados los siguientes recorridos Preorden y Postorden:

Pre = {A,Z,W,X,Y,T,R,Q,V,L}.

Post = {Z,W,X,R,Y,Q, V,L,T,A}.

(2-a) Hay exactamente 2 árboles binarios con esos recorridos

(2-b) No hay ningún árbol binario con esos recorridos

(2-c) Hay exactamente 1 árbol binario con esos recorridos

(2-d) Hay más de 2 árboles binarios con esos recorridos

Razona la respuesta

- (3)** Supongamos que hacemos las 3 siguientes afirmaciones:

(a) En un esquema de hashing **puede usarse como función hash** $h(x)=M - (x \% M)$ con M primo.

(b) Si insertamos un **conjunto de enteros ordenados** en un **AVL** y en un **APO**, la **altura** de ambos es la misma.

(c) El **orden relativo en que las hojas** se listan en los recorridos preorden, inorden y postorden de un **APO** es el mismo en los 3 casos

3-a: Las tres son ciertas **3-b:** Dos son ciertas y una falsa **3-c:** Dos son falsas y una cierta;

3-d: Las tres son falsas.

Razona la respuesta.

2. (1.5 puntos) Se define una **matriz singular** como una **matriz nxn** en la que se van alternando elementos pares e impares en círculos (siempre el mismo valor en cada caso). Un ejemplo de este tipo de matrices es la matriz que se muestra.

2	2	2	2	2	2
2	5	5	5	5	2
2	5	8	8	5	2
2	5	8	8	5	2
2	5	5	5	5	2
2	2	2	2	2	2

Proponer una representación eficiente (basada en el tipo vector<list>) para la matriz e implementar:

a) el **operator()** que devuelva el elemento en la fila fil, columna col.

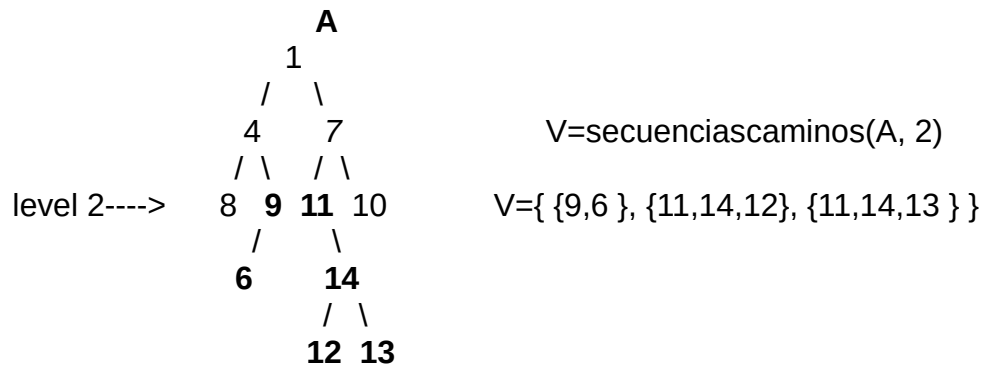
un iterador que itere sobre los elementos impares de la matriz. Han de implementarse (aparte de las de la clase iteradora) las funciones **begin()** y **end()**.



3. (1.5 puntos) Dado un árbol binario de enteros A y un nivel en el árbol level, implementar una función que encuentre los caminos de longitud mayor o igual que uno desde cualquier nodo de ese nivel a una hoja en el árbol. Devolver la solución en un vector de listas.

vector<list> secuenciascaminos (bintree<int> A, int level);

Ejemplo:



4. (1.5 puntos) Dada una lista de conjuntos **list< set<int> > L**, implementar una función

bool proper_subset(list< set<int> > &L);

que devuelve true si los conjuntos de la lista L son subconjuntos propios en forma consecutiva. Es decir, si $L = (A_0, A_1, \dots, A_{n-1})$, debe ocurrir que $A_j \subset A_{j+1}$ para $j = 0, \dots, n-2$. ($A \subset B$ indica inclusión propia, es decir $A \subseteq B$ y $A \neq B$.)

5. (0.75 puntos) Insertar en el orden indicado (detallando los pasos) las siguientes claves en un **AVL**: {56, 37, 58, 72, 57, 80, 38, 33, 30}

(b) Insertar los elementos anteriores en el orden indicado (detallando los pasos) en un **APO**. **Borrar un elemento**

Tiempo: 2.30 horas