

Parte A

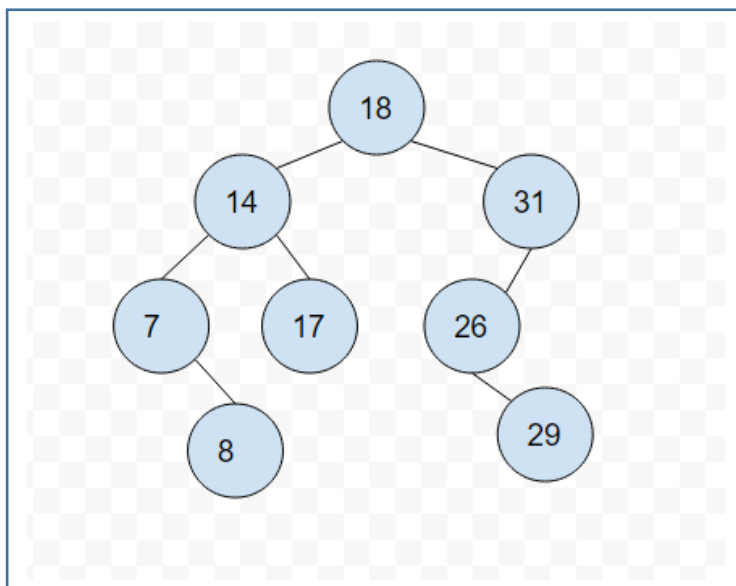
1. Implemente el tipo de datos abstracto ColaTDA utilizando solamente dos pilas como únicas estructuras auxiliares (no se pueden definir en su implementación ni otras estructuras ni variables enteras.)
2. Determine los costos de sus métodos.
3. Determinar los costos de las operaciones básicas del TDA GrafoImpl.

Parte B

1. Dado dos objetos de tipo **GrafoTDA**, **G1**, **G2**, de enteros, generar un **conjuntoTDA** correspondiente a los vértices de igual etiqueta que tengan igual promedio de pesos de las aristas salientes e igual promedio de pesos de aristas entrantes.
2. Dado un **ABBTDA A** y un **grafoTDA G**, generar un conjunto que contenga todos los nodos del árbol cuyo valor sea igual a la suma de pesos de las aristas salientes y entrantes de alguno de los vértices del grafo.

Parte C

1. Dado el siguiente árbol:
 - a) Determinar si es AVL, justificar la respuesta
 - b) Si no es un árbol AVL, explique por qué no lo es y realice las rotaciones necesarias para que lo sea. Detalle cada paso.



Programación II
Final regular

<pre>public interface PilaTDA { void apilar(int x); void desapilar(); int tope(); boolean pilaVacía(); void inicializarPila(); }</pre>	<pre>public interface ColaPrioridadTDA{ void acolarPrioridad(int x, int prior); void desacolar(); int primero(); int prioridad(); boolean colaVacía(); void inicializarCola(); }</pre>	<pre>public interface ABBTDA { int raiz(); ABBTDA hijoIzq(); ABBTDA hijoDer(); boolean arbolVacio(); void inicializarArbol(); void agregarElem(int x); void EliminarElem(int x); }</pre>
<pre>public interface ColaTDA{ void acolar(int x); void desacolar(); int primero(); boolean colaVacía(); void inicializarCola(); }</pre>	<pre>public interface DiccionarioSimpleTDA{ void inicializarDiccionario(); void agregar(int clave, int valor); void eliminar(int clave); int recuperar(int clave); ConjuntoTDA claves(); }</pre>	<pre>public interface GrafoTDA { void inicializarGrafo(); void agregarVertice(int v); void eliminarVertice(int v); ConjuntoTDA vertices(); void agregarArista(int v1, int v2, int p); void EliminarArista(int v1, int v2); boolean ExisteArista(int v1, int v2); int PesoArista(int v1 , int v2); }</pre>
<pre>public interface ConjuntoTDA{ void agregar(int x); void sacar(int x); boolean pertenece(int x); int elegir(); boolean conjuntoVacio(); void inicializarConjunto(); }</pre>	<pre>public interface DiccionarioMultipleTDA{ void inicializarDiccionario(); void agregar(int clave, int valor); void eliminar(int clave); void eliminarValor(int cl, int valor); ConjuntoTDA recuperar(int clave); ConjuntoTDA claves(); }</pre>	