



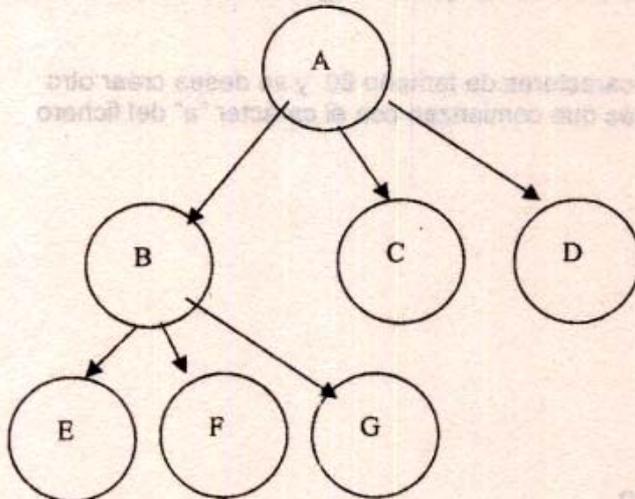
Curso : Examen Final Estructura de Datos y Algoritmos II
Grupo :
Profesor(es) : Javier Cabrera Díaz y Sofía Álvarez Cárdenas
Fecha : 1 de Diciembre
Hora : 6:30pm a 8:30pm
Duración de la prueba : 2 horas

Nota: El examen es sin copias ni apuntes.
Está prohibido el préstamo de calculadoras, correctores, uso de celulares, consumo de bebidas, comidas y cigarrillos.

Se recomienda contestar las preguntas en el mismo orden en que aparecen
Utilice las clases cuya definición se anexa. No tiene que completar sus métodos, solo realice los métodos para responder las preguntas que se realizan

I. Parte Práctica

Dado el siguiente árbol:

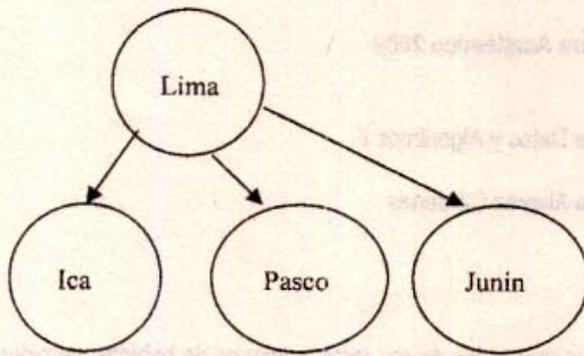


1. Represente Gráficamente el árbol binario correspondiente (1 punto)
2. Obtenga los nodos en el recorrido de orden final (1 punto)
3. Representélo secuencialmente en orden familiar (2 punto)
4. Responda: ¿Puede representarse como un árbol hilvanado? Justifique su respuesta (2 punto)

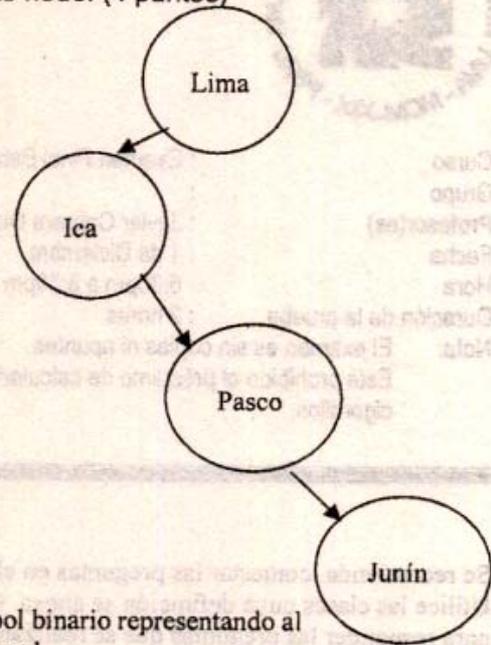
II Parte Práctica

Realice los métodos en Java. Puede haber errores sintácticos o algunos elementos en pseudo código.

1. Se tiene un árbol general, representado por un árbol binario, en la que cada nodo representa una ciudad y sus hijos a las ciudades a las que se puede llegar sin pasar por otra. Se desea **dada la referencia a un nodo** (o dirección o puntero) obtener las ciudades a las que se puede llegar desde este nodo. (4 puntos)



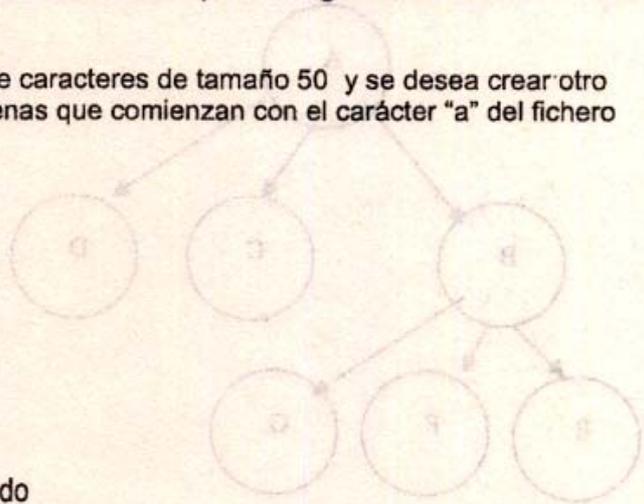
Árbol general



Árbol binario representando al general

En el ejemplo desde Lima se puede llegar a Ica, Pasco y Junin

2. Se tiene un grafo y se conoce **la llave de un vértice** de este y se desea construir un programa para eliminar todas las aristas de las que es **origen o destino** dicho vértice (6 puntos)
3. Se tiene un fichero de cadenas de caracteres de tamaño 50 y se desea crear otro fichero donde solo estén las cadenas que comienzan con el carácter "a" del fichero original (4 puntos)



Anexo

Para árbol binario

```

public class Nodo {
    private int llave; //la llave del nodo
    private String inf; //la informacion del nodo
    private Nodo hijoIzq; //el hijo izquierdo del nodo
    private Nodo hijoDer; //el hijo derecho del nodo
    public Nodo(int id, String cc)//Constructor
    {}
    public void mostrar()
    {
        System.out.print ("");
        System.out.print (llave);
        System.out.print ("");
        System.out.print (inf);
        System.out.println ("");
    }
}
  
```

```

public Nodo dameHijoIzq () { }
public Nodo dameHijoDer () { }
    public int dameLLave () { }
    public String dameInf () { }
public void ponerHijoIzq (Nodo hijo) { }
    public void ponerHijoDer (Nodo hijo) { }
}
public class Arbol {
    public Nodo raiz;
    public Arbol() {
        raiz =null;
    }
}

```

Sobre grafo

```

public class Vertice
{ private String llave;
  private boolean fueVisitado;
  public Vertice(String llav)
  {
  }
  public String dameLlave()
  {
  }

  public boolean yaFueVisitado()
  {
  }
} //fin de clase vértice

public class AdyMatriz {

    private int Max_Verts =20;
    private Vertice verticeLista[]; //arreglo de vertices
    private int adyMat[][]; //matriz de adyacencia
    private int nVertices; //numero actual de vertices

    public AdyMatriz()
    {
    }
    public void insertarVertice(String llav)
    {
    }

    public void insertarArco(int inicio, int fin)
    {
    }

    public void muestraVertice(int v)

```

```
{  
}  
public void muestraMatriz()  
{  
}
```

Sobre ficheros

```
File(String S)
```

```
File f= new File("MiArchivo");
```

```
File f1,f2,f3;
```

```
f1 = new File("Pedro");
```

```
f2 = new File("Juan");
```

```
f3 = new File("Diego");
```

```
f1.delete();
```

```
f2.renameTo(f3);
```

```
FileOutputStream fs = null;
```

```
fs = new FileOutputStream("texto.txt");
```

```
fs.write(buffer,0,nbytes);
```

```
FileInputStream fe = null;
```

```
fe = new FileInputStream("texto.txt");
```

```
nbytes = fe.read(buffer,0,81
```

Nota: El martes 5 a las 2pm será la devolución del examen en sala de profesores B203 o en la Escuela de Ingeniería Informática B206