**Universidad Interamericana de Panamá**

**Facultad de Ingeniería, arquitectura y diseño**

**Escuela de Ingeniería y Sistemas.**

Materia: Estructura de Datos II Cuatrimestre: IIQ\_2019 Código: 301-00040 Facilitador: Leonardo Esqueda

Nombre completo: \_\_\_Zulendis De Gracia\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID: \_\_\_9-745-165\_\_\_\_\_\_\_

Carrera: \_\_Ingenieria industrial y de Sistemas\_\_\_\_\_\_\_

¿Qué son las Estructuras de Datos? 5ptos

R= Las estructuras de datos en programación son diferentes formas de organizar información para manipular, buscar e insertar estos datos de manera eficiente.

2. ¿De dónde vienen las estructuras de datos? 5ptos

* Dispositivos
* supermercados

3. ¿Mencione las dos (2) ramas de la estructura de datos interna? 5ptos

* Estática
* dinámica

4. ¿Cómo interactúan las estructuras de datos internas y externas? 5ptos

5. ¿Cómo funcionan los árboles Binarios? 5ptos

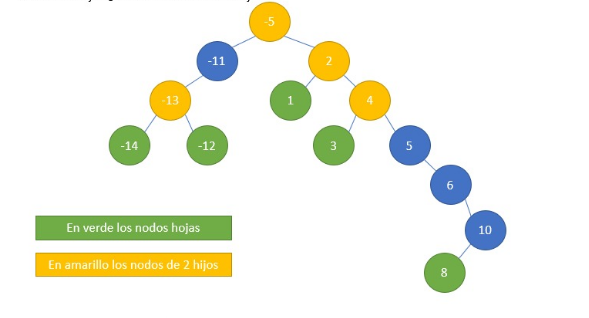
El árbol binario es el caso más simple de árbol de orden N, cuando N vale 2. Su especificación se puede hacer considerando un valor constante, el árbol nulo, y un constructor de árboles a partir de un elemento y dos árboles.

6. Describa un caso en donde usted podría realizar y utilizar un árbol binario. 5ptos

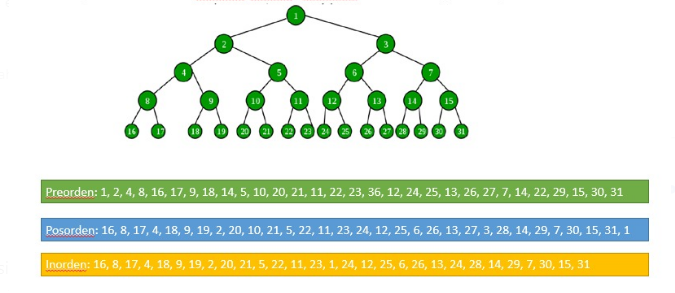
Motor de búsqueda de información por ejemplo, Google, opera

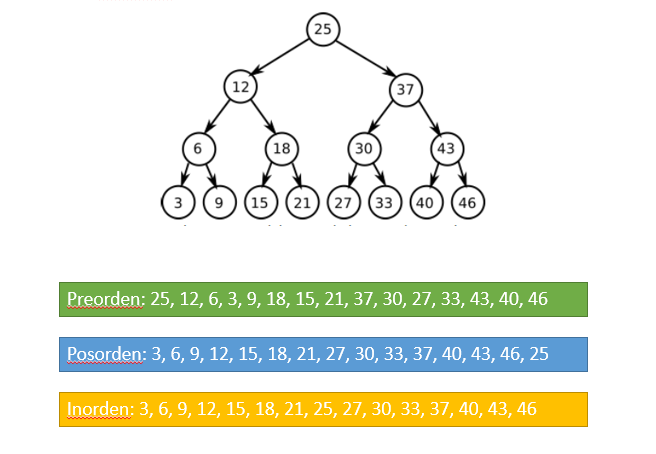
7. Mencione y explique 5 propiedades de los métodos de ordenamiento de los arboles binarios. 5ptos

8. Distribuya los nodos -5,2,-11,4,-13,5,3,-14,1,6,10,-12,8 en un árbol binario y determine ¿Cuáles son los nodos hoja? ¿Cuáles son nodos de 2 hijos? 5ptos



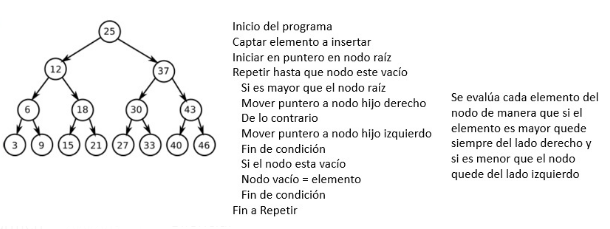
9. Realice el recorrido preorden, inorden y postorden de los árboles siguientes: 5ptos





10. Describa un problema del mundo real y que usted considere que pueda ser solucionado utilizando las estructuras de datos. Mencione qué modelo estructura de datos utilizaría, porqué y desarrolle la solución 10ptos

Práctica:

1. Realice un algoritmo lógico el cual explique y demuestre el proceso de inserción de los elementos del problema teórico número dos (2) en un árbol binario. 15ptos
2. 

2. Desarrolle un programa en el lenguaje C++ o python en donde cargue por defecto los elementos del árbol: 9 – A y realice los recorridos: preorden, inorden y postorden 15ptos

#Clase llamada nodo

class Nodo:

#Constructor con los argumentos

def init(self,numero, izq=None, der=None):

#Punteros

self.numero = numero

self.izq = izq

self.der = der

#

# #Retornar el valor

# def str(self):

# return self.poss

#Clase del arbol binario

class aBinarios:

#metodo constructor y el atributo raiz|

def init(self):

self.raiz = None

#creamos la funcion agregar

def agregar(self):

#Comparamos si la raiz esta vacia, de ser cierto añadiremos el nodo en la Raiz

padre = self.raiz

self.raiz = nodo(1)

For i=2 in Range(32):

if padre.izq = None:

padre.izq= nodo(i)

else:

if padre.der == None

padre.der= nodo(i)

else:

if (padre.izq != None) && (padre.izq == None)

padre = padre.izq

else:

if (padre.izq != None) && (padre.izq != None)

padre = padre.der

def inorder(self, a):

if a == None:

return None

else:

self.inorder(a.left)

print(a.dato)

self.inorder(a.right)

def preorder(self, a):

if a == None:

return None

else:

print(a.dato)

self.preorder(a.left)

self.preorder(a.right)

def postorder(self, a):

if a == None:

return None

else:

self.postorder(a.left)

self.postorder(a.right)

print(a.dato)

#se crea funcion que permite obtener la raiz

def getRaiz(self):

return self.raiz

#creamos un menu

if name == "main":

#Creamos una instancia de la Clase aBinarios

ab = aBinarios()

poss=0

while(True):

#opciones del menu

print("Arboles\_Binarios\n"+

"1. Agregar Usuario Nuevo\n"+

"2. Buscar Usuario\n"+

"3. Salir\n")

num = input("Ingrese la opcion: ")

if num == "1":

ab.agregar(nod)

elif num == "2":

nombre= input("Ingrese el nombre del Usuario: ")

print("El Usuario se encuentra en la posicion: ",ab.Buscar(nombre,ab.getRaiz()))

elif num == "3":

exit()

10. Las Torres de Hanoi es un juego matemático que consiste en tres varillas verticales y un número indeterminado de discos que determinarán la complejidad de la solución. No hay dos discos iguales, están colocados de mayor a menor en una varilla ascendentemente, y no se puede colocar ningún disco mayor sobre uno menor a él en ningún momento. El juego consiste en pasar todos los discos a otra varilla colocados de mayor a menor ascendentemente. Leyenda: Dios al crear el mundo, colocó tres varillas de diamante con 64 discos en la primera. También creó un monasterio con monjes, los cuales tienen la tarea de resolver esta Torre de Hanoi divina. El día que estos monjes consigan terminar el juego, el mundo acabará. El mínimo número de movimientos que se necesita para resolver este problema es de 264-1. Si los monjes hicieran un movimiento por segundo, los 64 discos estarían en la tercera varilla en poco menos de 585 mil millones de años. Como comparación para ver la magnitud de esta cifra, la Tierra tiene como 5 mil millones de años, y el Universo entre 15 y 20 mil millones de años de antigüedad, sólo una pequeña fracción de esa cifra. Resolución: el problema de las Torres de Hanoi es curioso porque su solución es muy rápida de calcular, pero el número de pasos para resolverlo crece exponencialmente conforme aumenta el número de discos. Para obtener la solución más corta, es necesario mover el disco más pequeño en todos los pasos impares, mientras que en los pasos pares sólo existe un movimiento posible que no lo incluye. El problema se reduce a decidir en cada paso impar a cuál de las dos pilas posibles se desplazará el disco pequeño: El algoritmo en cuestión depende del número de discos del problema. Si inicialmente se tiene un número impar de discos, el primer movimiento debe ser colocar el disco más pequeño en la pila destino, y en cada paso impar se le mueve a la siguiente pila a su izquierda (o a la pila destino, si está en la pila origen). La secuencia será DESTINO, AUXILIAR, ORIGEN, DESTINO, AUXILIAR, ORIGEN, etc. Si se tiene inicialmente un número par de discos, el primer movimiento debe ser colocar el disco más pequeño en la pila auxiliar, y en cada paso impar se le mueve a la siguiente pila a su derecha (o a la pila origen, si está en la pila destino). La secuencia será AUXILIAR, DESTINO, ORIGEN, AUXILIAR, DESTINO, ORIGEN, etc. 15 ptos