UNSA

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

Estructuras Discretas II

Docente: Carlo Corrales Delgado

Actividad

Ejercicios de Lección 8

Escuela:

Ciencia de la computación (Primer año)

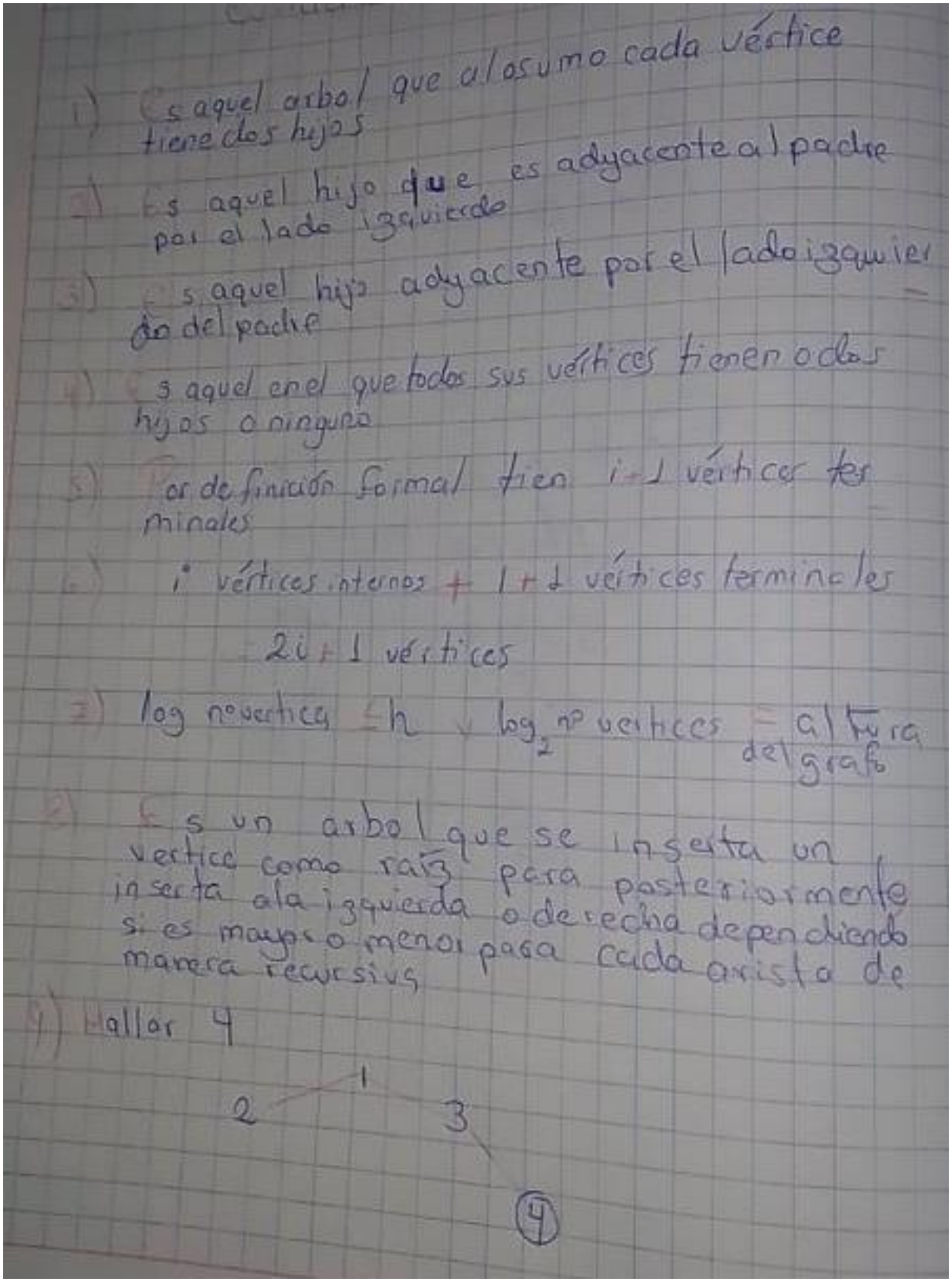
Temas:

-Árboles Binarios

Alumno:

Josue Gabriel Sumare Uscca

Ejercicios de repaso



1) Es aquel arbol que asumo cada vértice tiene dos hijos

2) Es aquel hijo que es adyacente al padre por el lado izquierdo

3) Es aquel hijo adyacente por el lado izquierdo del padre

4) Es aquel en el que todos sus vértices tienen o dos hijos o ninguno

5) Por definición formal tiene i+1 vértices terminales

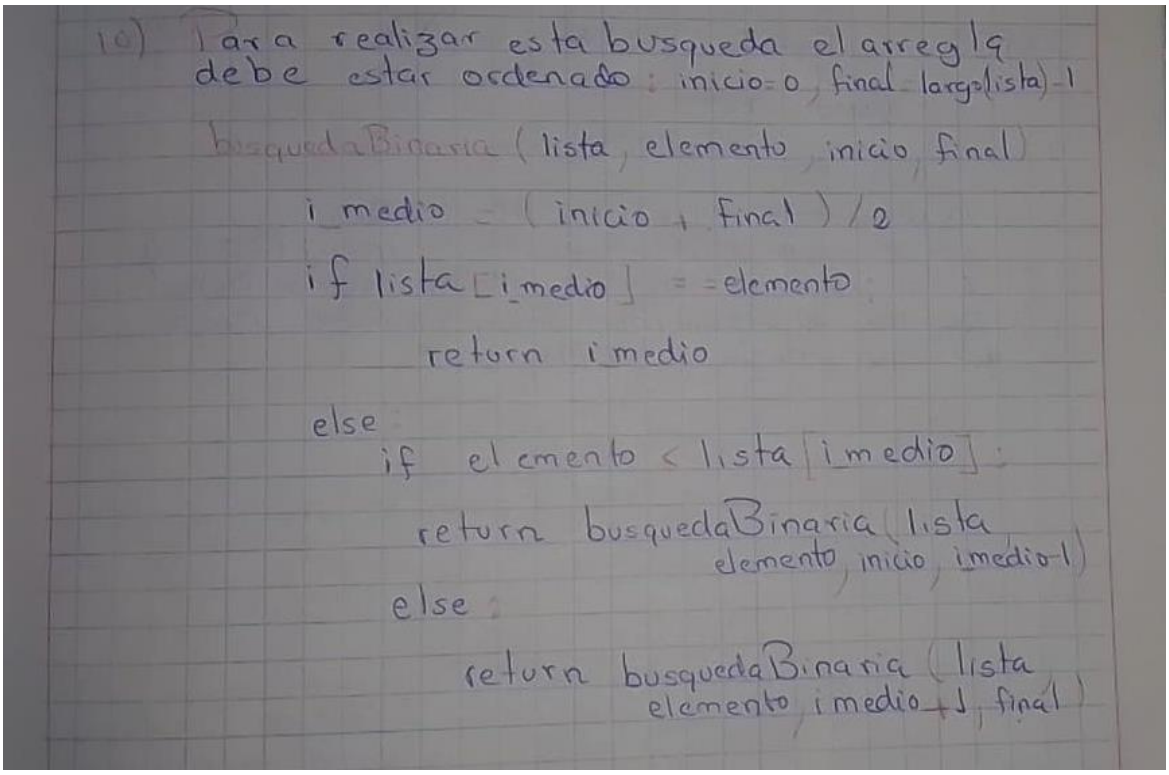
i° vértices internos + 1 + 1 vértices terminales

2i + 1 vértices

6) log nº vertices = h → $\log_2$ nº vertices = altura del grafo

7) Es un arbol que se inserta un vertice como raiz para posteriormente inserta ala izquierda o derecha dependiendo si es mayor o menor para cada arista de manera recursiva

8) Hallar 4


10) Para realizar esta busqueda el arreglo debe estar ordenado: inicio=0, final=largo(lista)-1
busquedaBinaria (lista, elemento, inicio, final)
i_medio = (inicio + final) / 2
if lista[i_medio] == elemento
return i_medio
else
if elemento < lista[i_medio]:
return busquedaBinaria (lista elemento, inicio, i_medio-1)
else:
return busquedaBinaria (lista elemento, i_medio+1, final)

Ejercicios

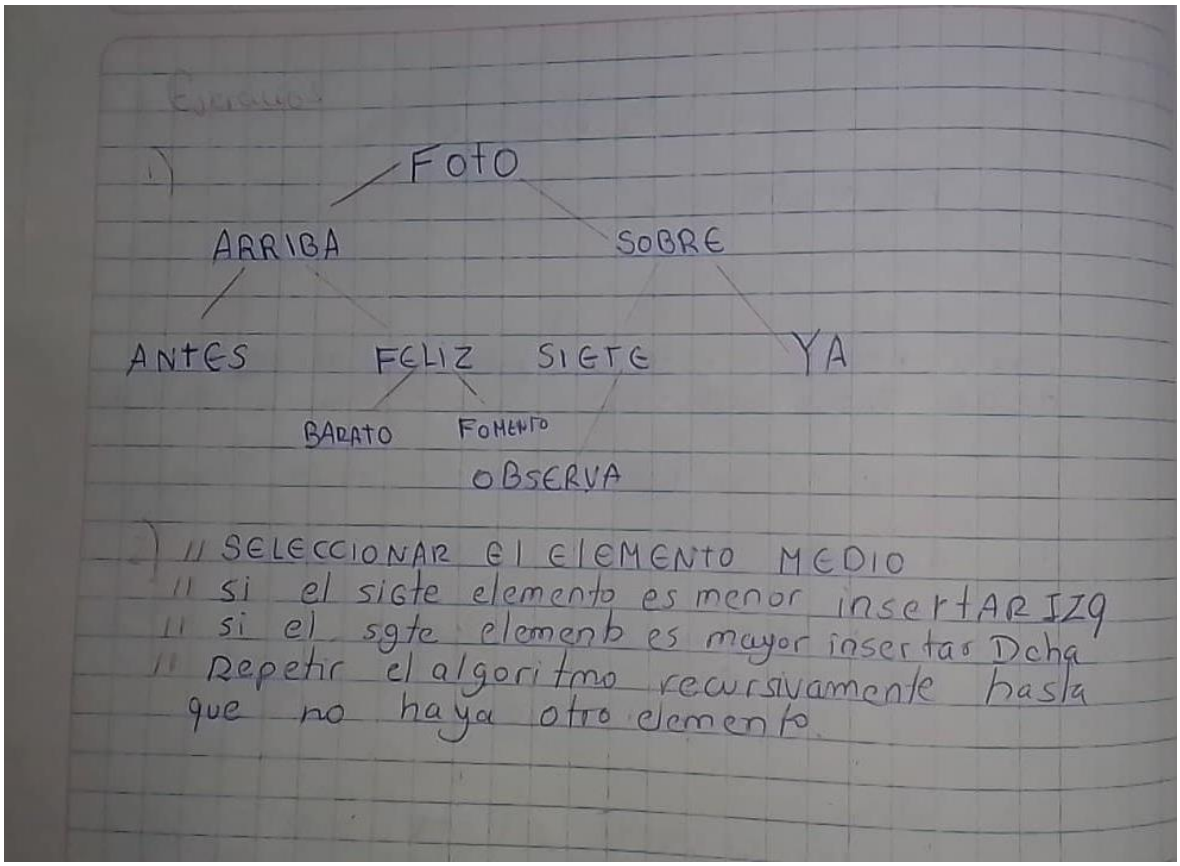
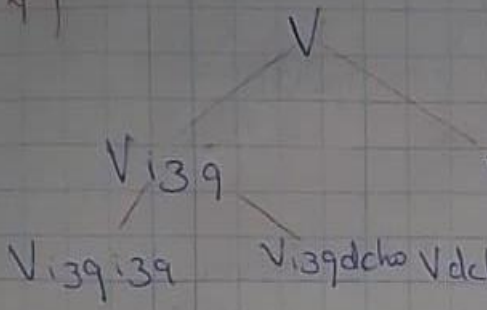


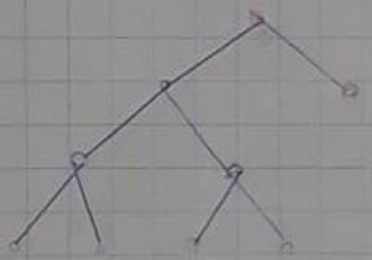
FOTO
ARRIBA
SOBRE
ANTES
FELIZ
SIETE
YA
BARATO
FOMENTO
OBSERVA
// SELECCIONAR EL ELEMENTO MEDIO
// si el sigte elemento es menor insertARIZQ
// si el sgte elemenb es mayor insertar Dcha
// Repetir el algoritmo recursivamente hasta que no haya otro elemento.

4)



- Si ademas de repetirse recursivamente en los hijos ordenando los menores que el padre en lado izquierdo y los mayores en el lado derecho este proceso nos permite realizar un algoritmo de busqueda de eficiencia O(log n).

5)



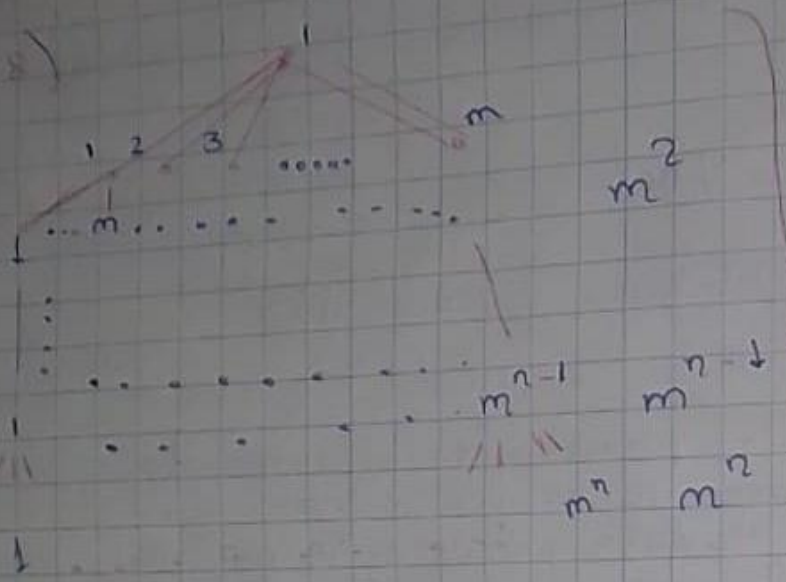
i vertices internos

i + 1 vertices terminales

Arbol binario completo

6) No se puede debido a que no se cumple la propiedad de la diferencia en 1 de hojas y vertices terminales

7) No se puede debido a que no se cumple la propiedad de la diferencia en 1 de hojas y vertices terminales



n altura

i: aristas internas

m^n arboles terminales

b) Por deducción

$$i = \frac{1 - m^n}{1 - m}$$ Aristas internas

m^n Aristas terminales

9) El algoritmo es el mismo

" Elegir un medio

" En base a esto iría evaluar si es menor o mayor y agregar a izquierda o derecha respectivamente

" Manejar en los niveles del árbol para seguir ordenando de esta manera los vértices hijos

10) // elegir un medio
// ordenar derecha o izq si es mayor o menor
// si es menor < medio
// medio = anterior al hijo izq + medio - 1

// retur AlgoritmoBinario (lista, elemento, inicio, imedio)
// de lo contrario AlgoritmoBinario (lista, elemento, imedio+1, final)

11) Altura = $\log_2$ n° vertices terminales

12) En la anterior llamada recursiva podemos incluir

// si largo (lista(inicio), lista(final)) > 3
retur n False
else:
return true

- El largo 3 se explica que para que sea binaria el subarbol debe estar conectado con a los umo un elemento menor y otro mayor, con lo que se contabilizaria 3 elementos

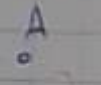
14) Si es balanceado ya que la diferencia de altura los subarboles a lo mucho da 1.

15) Si ya que las subgráficas tambien en cuestión de altura se diferencian en 1

16) Tambien es balanceado ya que las alturas de las subgraficas difieren en 0 y en 1

17) Tambien es balanceado porque los subgráficas posibles difieren en 1

18) $N_0 = 1$

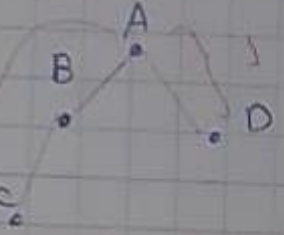


Verdad

$N_1 = 2$

Verdad

$N_2 = 4$



Verdad