

Tarea 1,2 Unidad 4.

Estructuras de Datos y algoritmos II.

Prof. Gerardo Tovar Tapia.

Alumna. Monroy Velázquez Alejandra Sarahí

Sección 1. Investigar, analizar, resolver.

Nota: Las actividades fueron hechas a mano y escaneadas.

- 1) Escribe en notación prefija y posfija las siguientes expresiones en notación infija.

$$(A+B)*(C-D)$$

$$A+B*C-D+E/F/(G+H)$$

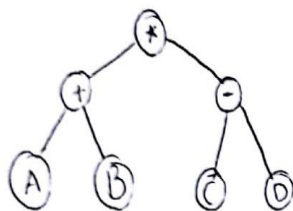
$$((A+B)*C-(D-E))/(F+G)$$

$$A-B/(C*D/E)$$

Nota: Hacerlo por el método de los árboles, y desarrollar todo el procedimiento.

• $(A+B)*(C-D)$

• Desarrollo en árbol



• Notación prefija:

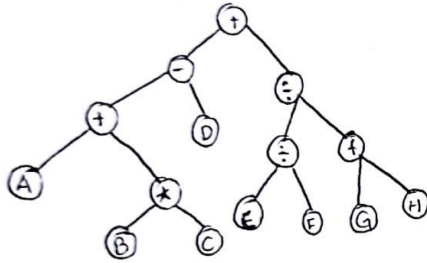
$*, +, A, B, -, C, D$

• Notación posfija:

$A, B, +, C, D, -, *$

• $A + B * C - D + E / F / (G + H)$

• Desarrollo en árbol



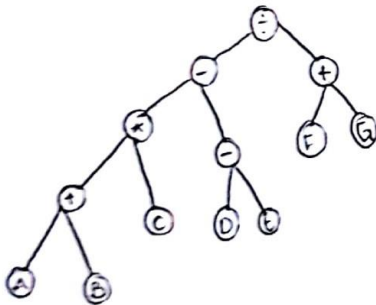
• Notación prefija

$+, -, +, A, *, B, C, D, \div, \div, E, F, +, G, H$

• Notación posfija

$B, C, *, A, +, D, -, E, F, \div, G, H, +, \div, +$

• $((A + B) * C - (D - E)) / (F + G)$



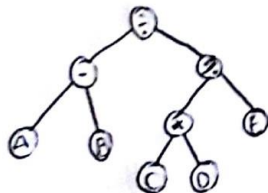
• Notación prefija:

$\div, -, *, +, A, B, C, -, D, E, +, F, G$

• Notación posfija:

$A, B, +, C, *, D, E, -, -, F, G, +, \div$

• $A - B / (C * D / E)$



• Desarrollo en árbol

• Notación prefija:

$\div, -, A, B, \div, *, C, D, E$

• Notación posfija:

$A, B, -, C, D, *, E, \div, \div$

2) Representa las expresiones aritméticas en notación posfija y prefija.

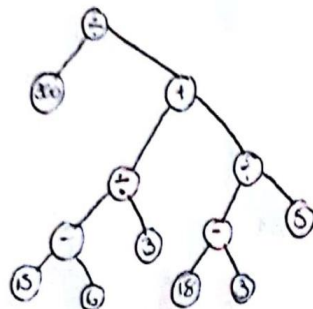
| | |
|---|--------|
| $300 \div [(15 - 6) \div 3 + (18 - 3) \div 5]$ | R. 50 |
| $9[15 \div (6 - 1) - (9 - 3) \div 2]$ | R. 0 |
| $[15 + (8 - 3)5] \div [(8 - 2) \div 2 + 7]$ | R. 4 |
| $(9 + 3)5 - 2 \div (3 - 2) + 8 \times 6 \div 4 \div 2 + 5$ | R. 69 |
| $[(9 - 4) \div 5 + (10 - 2) \div 4] + 9 \times 6 \div 18 + 2$ | R. 8 |
| $500 - \{(6 - 1)8 \div 4 \times 3 + 16 \div (10 - 2)\} - 5$ | R. 463 |

| | |
|-------------------------------|------------------------------------|
| $\frac{8x^2 + 16x}{8x}$ | $\frac{2w^3 - 4w^2}{4w^2}$ |
| $\frac{y^2 - 25}{5 - y}$ | $\frac{16 - b^2}{b - 4}$ |
| $\frac{1 - c^2}{c - 1}$ | $\frac{m^2 - 25}{m^2 - 9m + 20}$ |
| $\frac{a^2 - b^2}{b^2 - a^2}$ | $\frac{8x^2 - 18y^2}{(2x + 3y)^2}$ |

Nota: Hacerlo por el método de los árboles, y desarrollar todo el procedimiento.

• $300 \div [(15 - 6) \div 3 + (18 - 3) \div 5]$

• Desarrolla en árbol



• Notación prefija

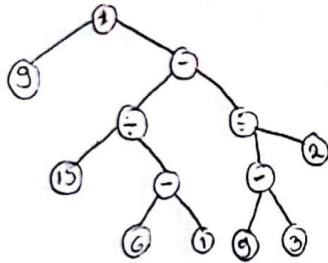
$\div, 300, +, \div, -, 15, 6, 3, \div, -, 18, 3, 5$

• Notación posfija

$300, 15, 6, -, 3, \div, 18, 3, -, 5, \div, +, \div$

• $9[15 \div (6-1) - (9-3) \div 2]$

• Desarrollo en árbol



• Notación prefija

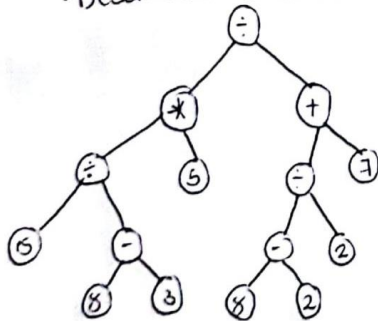
$\times, 9, -, \div, 15, -, 6, 1, \div, -, 9, 3, 2,$

• Notación posfija

$9, 15, 6, 1, -, \div, 9, 3, -, 2, \div, -, \times$

• $[15 \div (8-3) 5] \div [(8-2) \div 2 + 7]$

• Desarrollo en árbol



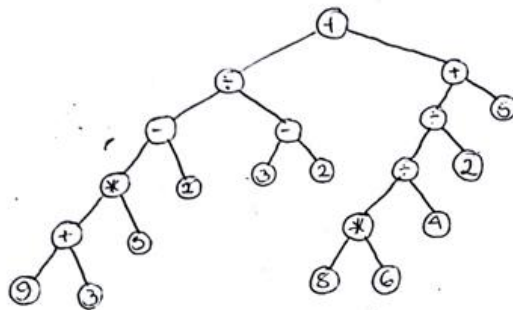
• Notación prefija

$\div, \times, \div, 15, -, 8, 3, 5, +, \div, -, 8, 2, 2, 7,$

• Notación posfija

$8, 3, -, 15, \div, 5, \times, 8, 2, -, 2, \div, 7, +, \div$

• $(9+3) 5 - 2 \div (3-2) + 8 \times 6 \div 4 \div 2 + 5$



• Desarrollo en árbol

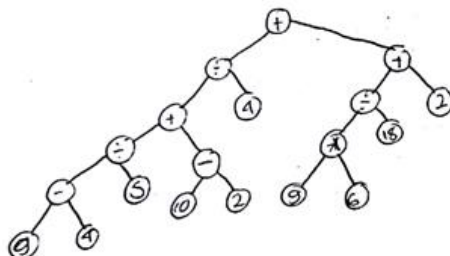
• Notación prefija:

$+, \div, -, \times, +, 9, 3, 5, 2, -, 3, 2,$
 $+, \div, \div, \times, 8, 6, 4, 2, 5.$

• Notación posfija:

$9, 3, +, 5, \times, 2, -, 3, 2, -, \div,$
 $8, 6, \times, 4, \div, 2, \div, 5, +, +.$

• $[(9-4) \div 5 + (10-2) \div 4] + 9 \times 6 \div 18 + 2$



• Desarrollo en árbol

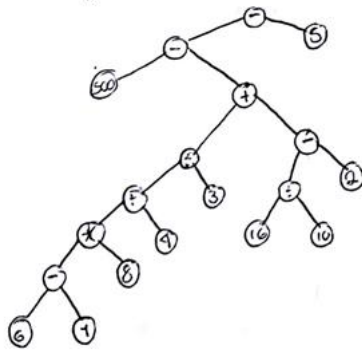
• Notación prefija:

$+, \div, +, \div, -, 9, 4, 5, -, 10, 2, 4, +, \div, \times,$
 $9, 6, 18, 2$

• Notación posfija:

$9, 4, -, 5, \div, 10, 2, -, +, 4, \div, 9, 6, \times, 18,$
 $\div, 2, +, +.$

• $500 - \{(6-1)8 \div 4 \times 3 + 16 \div 10 - 2\} - 5$



• Desarrollo en árbol

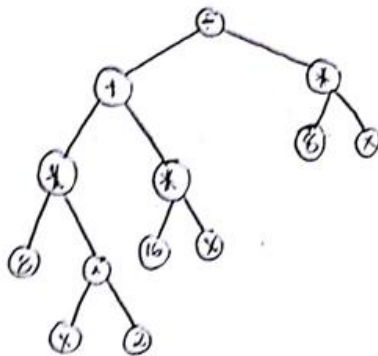
• Notación prefija:

$- , - , 500 , + , \div , \div , * , - , 6 , 1 , 8 ,$
 $1 , 3 , - , \div , 16 , 10 , 2 , 5 ,$

• Notación posfija:

$500 , - , 6 , 1 , - , 8 , * , 4 , \div , 3 , \div , 16 , 10 , \div , 2 , - ,$
 $+ , 5 , - .$

• $\frac{8x^2 + 16x}{8x}$



• Desarrollo en árbol

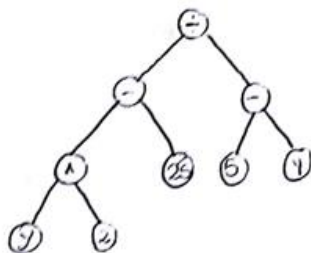
• Notación Prefija:

$\div , + , * , 8 , ^ , x , 2 , * , 16 , x , 8 , x$

• Notación posfija:

$8 , x , 2 , ^ , * , 16 , x , * , + , 8 , x , * , \div$

• $\frac{y^2 - 25}{5 - y}$



• Desarrollo en árbol

• Notación Prefija

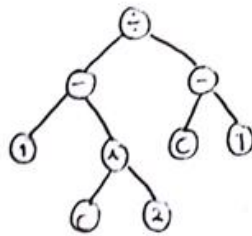
$\div , - , ^ , y , 2 , 25 , - , 5 , y$

• Notación posfija

$y , 2 , ^ , 25 , - , 5 , y , - , \div$

• Desarrollo en árbol

$$\frac{1-c^2}{c-1}$$



• Notación prefija

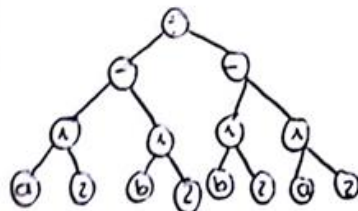
÷, -, 1, ^, c, 2, -, c, 1

• Notación posfija

1, c, 2, ^, -, c, 1, -, ÷

• Desarrollo en árbol

$$\frac{a^2-b^2}{b^2-a^2}$$



• Notación prefija

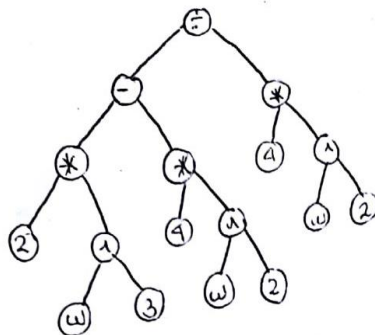
÷, -, ^, a, 2, ^, b, 2, -, ^, b, 2, ^, a, 2

• Notación posfija

a, 2, ^, b, 2, ^, -, b, 2, ^, a, 2, ^, -, ÷

• Desarrollo en árbol

$$\frac{2w^3-4w^2}{4w^2}$$



• Notación prefija

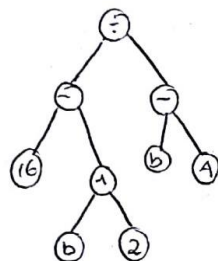
÷, -, *, 2, ^, w, 3, *, 4, ^, w, 2, ^, w, 2

• Notación posfija

2, w, 3, ^, *, w, 2, ^, 4, *, w, 2, ^, w, 2, ^, ÷

• Desarrollo en árbol

$$\frac{16-b^2}{b-4}$$



• Notación prefija

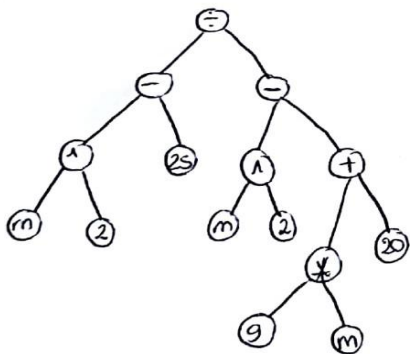
÷, -, 16, ^, b, 2, -, b, 4

• Notación posfija

b, 2, ^, 16, -, b, 4, -, ÷

• Desarrollo en árbol

$$\frac{m^2 - 25}{m^2 - 9m + 20}$$



Notación prefija

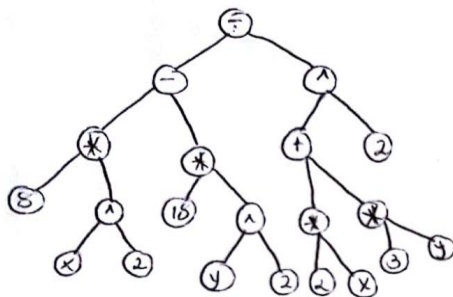
÷, -, ^, m, 2, 25, -, ^, m, 2, +, *, g, m, 20

Notación posfija:

m, 2, ^, 25, -, g, m, *, 20, +, m, 2, ^, -, ÷

Desarrollo en árbol

$$\frac{8x^2 - 18x^2}{(2x + 3y)^2}$$



Notación prefija

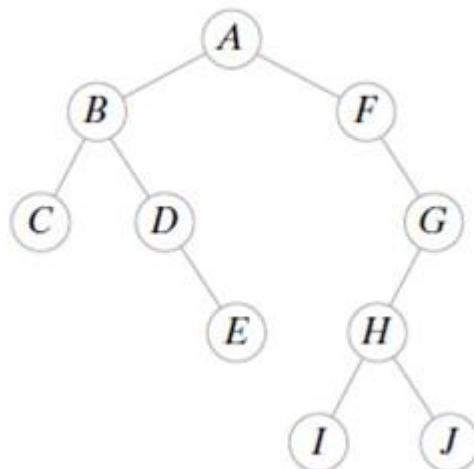
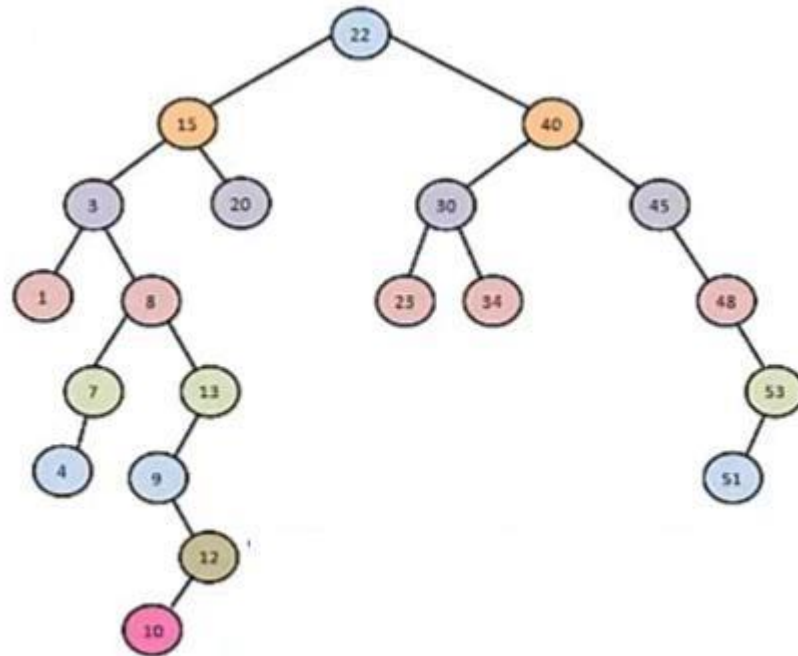
÷, -, *, 8, ^, x, 2, *, 18, ^, y, 2, x, +, *, 2, x, *, 3, y, 2

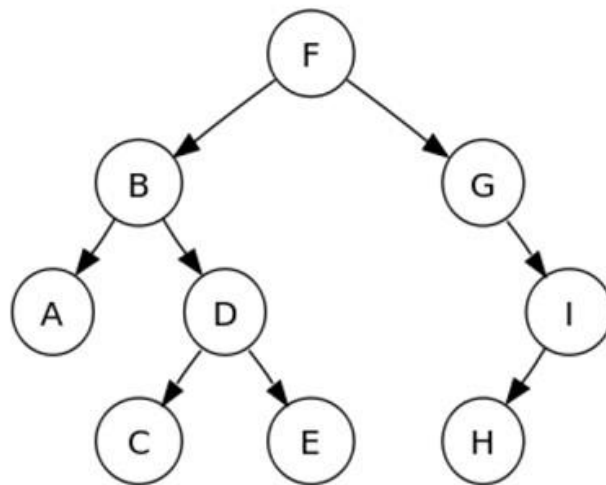
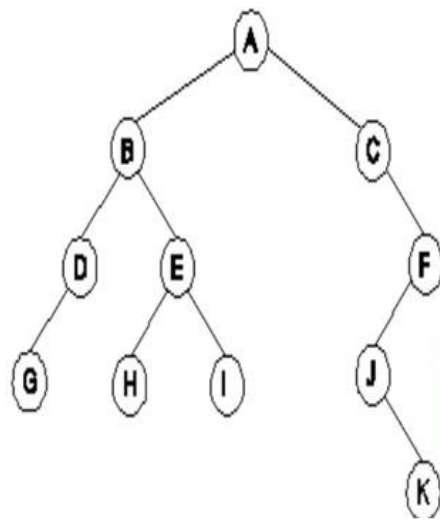
Notación posfija

x, 2, ^, 8, *, y, 2, ^, 18, *, 2, x, *, 3, y, *, +, 2, ^, ÷

Desarrollo en árbol

3) Dar el recorrido de los siguientes árboles en inorden, postorden, preorden.





a) • INORDEN - 10, 12, 9, 13, 8, 1, 7, 1, 3, 20, 15, 22, 51, 53, 18, 45, 10, 34, 25, 30

• POSTORDEN - 10, 12, 9, 13, 1, 7, 8, 1, 3, 20, 15, 51, 53, 18, 45, 34, 25, 30, 10, 22

• PREORDEN - 22, 15, 3, 1, 8, 7, 4, 13, 9, 12, 10, 20, 40, 30, 25, 34, 45, 18, 53, 51

b) • INORDEN - C, D, C, A, J, H, J, G, F

• POSTORDEN - C, D, C, B, J, J, H, G, F, A

• PREORDEN - A, B, C, D, E, J, J, H, G, F

c) • INORDEN - G, D, B, H, E, I, A, K, J, F, C

• POSTORDEN - G, D, H, I, E, B, K, J, F, C, A

• PREORDEN - A, B, D, G, E, H, I, K, J, F, G

d) • INORDEN - A, B, C, D, E, F, H, J, G

• POSTORDEN - A, C, E, D, B, H, J, G, F

• PREORDEN - F, B, A, C, E, D, H, J, G

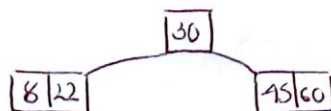
- 4) Dibujar el Árbol B a partir de los siguientes datos en Streaming, iniciando de lado derecho, el árbol debe ser de grado 4.

30,60,45,8,22,35,4,28,52,33,13,39,41,43,24,25,15

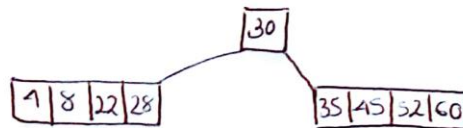
Nota: Hacer el desarrollo completo.

8 30 45 60

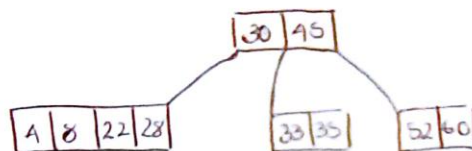
Insertamos 22



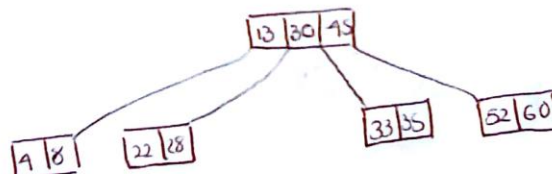
Insertamos 35, 4, 28, 52



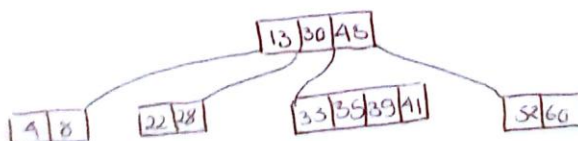
Insertamos 33



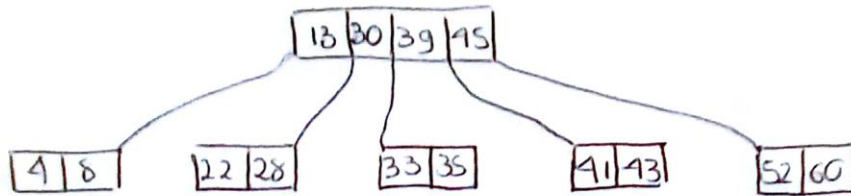
Insertamos 13



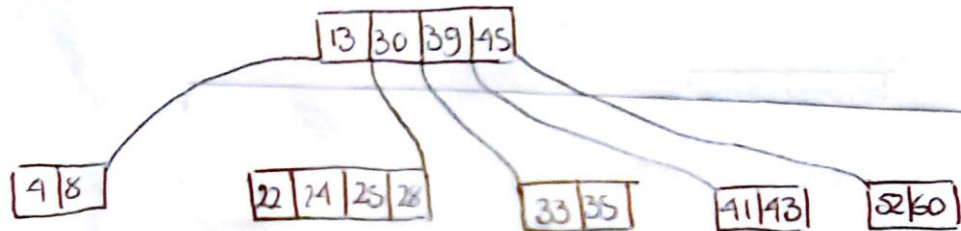
Insertamos 39, 41



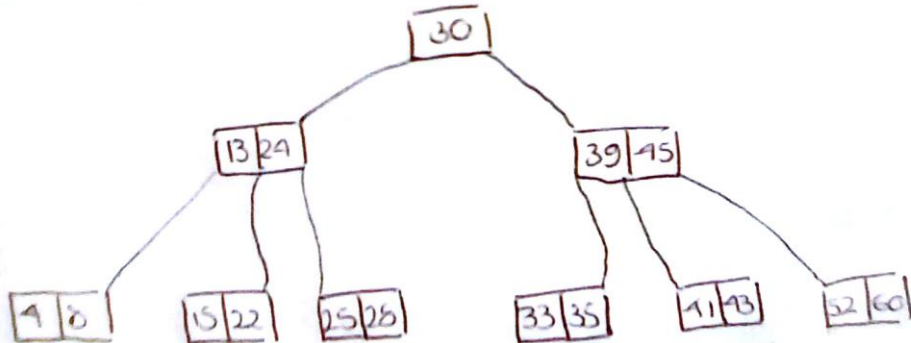
• Insertamos 43



• Insertamos 24, 25



• Insertamos 15



- 5) Dado el siguiente árbol B, con las características descritas, insertar los siguientes valores. (7,19,199, 100, 70, 50).

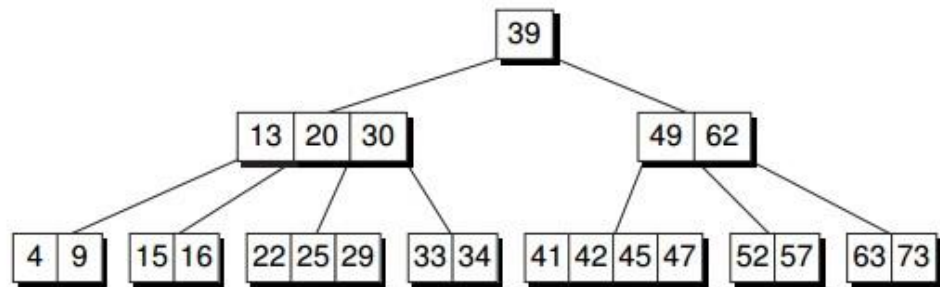
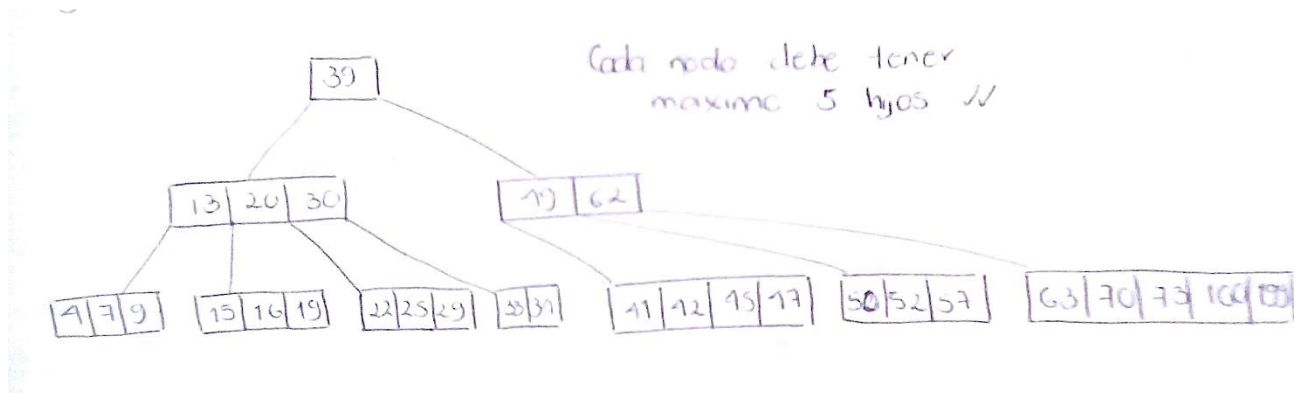


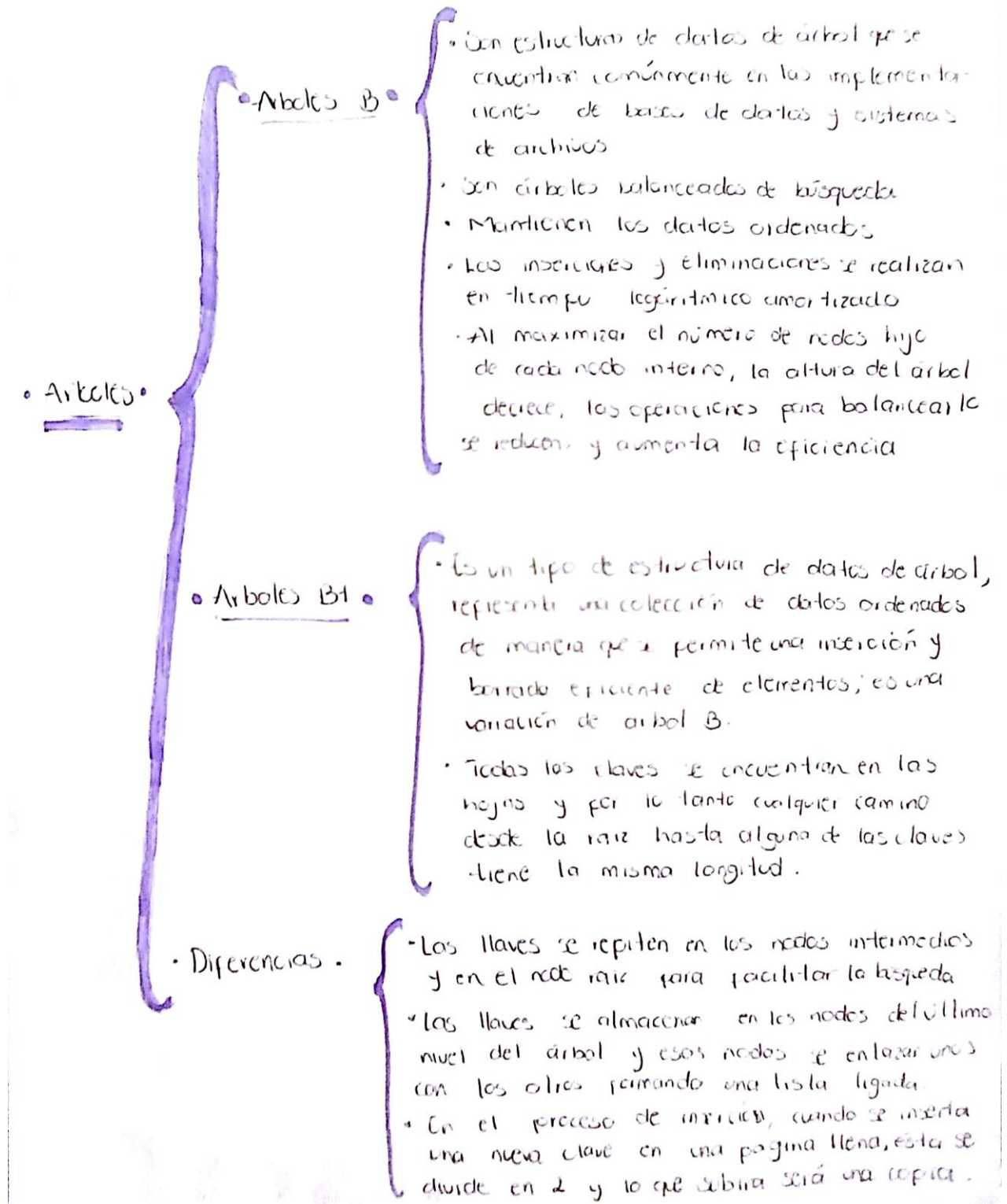
Figura 4.24: Ejemplo de árbol B de orden $p = 5$. Cada nodo interno excepto la raíz, debe tener entre 3 y 5 hijos o, equivalentemente, entre 2 y 4 entradas.



- 6) Dibujar un árbol B, de grado 3, dados los siguientes datos en Streaming.
(1,3,5,7,9,22,45,67,100,46, 88,200,201,20,39,70)

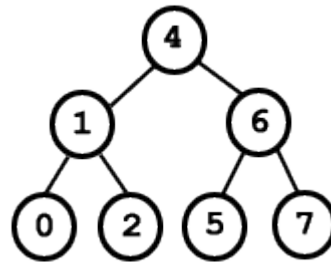
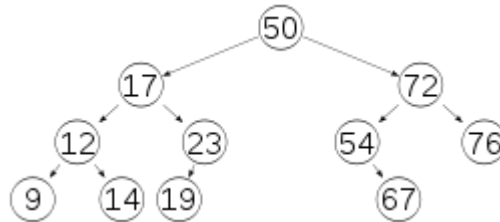
Nota: Hacer el desarrollo completo.

7) Investigar que es un árbol B+, hacer un cuadro sinóptico con sus principales características, y explicar diferencias y similitudes con los árboles b.



8) Investigar que es un árbol AVL, y dar 2 ejemplos.

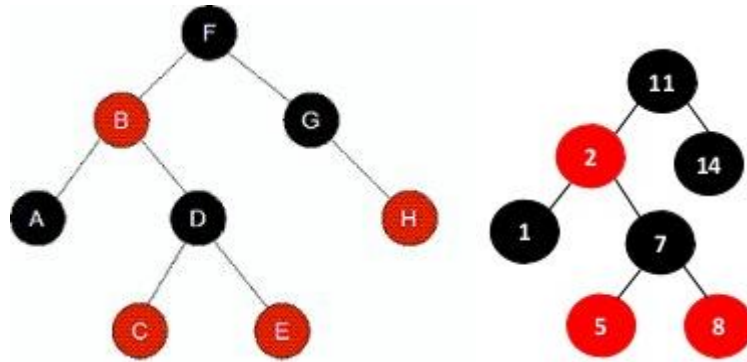
Los árboles AVL están siempre equilibrados, de tal modo que para todos los nodos, la altura de la rama izquierda no difiere en más de una unidad a la rama derecha o viceversa.



9) Investigar que es un árbol Rojo – Negro. Y dar dos ejemplos.

Un árbol rojo-negro es un árbol binario de búsqueda en el que cada nodo tiene un atributo de color cuyo valor es rojo o negro. Además de los requisitos impuestos a los árboles binarios de búsqueda convencionales, se deben satisfacer las siguientes reglas:

- Todo nodo es bien rojo o bien negro.
- La raíz es negra.
- Todas las hojas (null) son negras.
- Todo nodo rojo debe tener 2 hijos negros.
- Cada camino desde un nodo dado a sus hojas descendientes contiene el mismo número de nodos negros.



Sección 2. Programar

10) Del libro “python para informáticos”. Leer el capítulo 9, hacer los ejercicios que se proponen en de dichas sección.

Ejercicio 9.1 Escribe un programa que lea las palabras de words.txt y las almacene como claves en un diccionario. No importa qué valores tengan. Después puedes usar el operador in como un modo rápido de comprobar si una cadena está en el diccionario.

```
Ejercicio9.1.py x
1 file=open("words.txt")
2 diccionario=dict()
3 cont=0
4 for linea in file:
5     cadena=linea.split()
6     for i in cadena:
7         if i not in diccionario:
8             diccionario[i]=i
9 print(diccionario)
10 bus=input("Introduce la llave a la que quieres acceder al valor: ")
11 if bus in diccionario:
12     print("El valor es:",diccionario[bus])
13 else:print("No se encontro la llave")
```

El programa primero imprime el diccionario con cada pareja valor-llave.

```
Símbolo del sistema
C:\Users\Alexandra\Downloads>python Ejercicio9.1.py
{'Writing': 'Writing', 'programs': 'programs', 'or': 'or', 'programming': 'progr
amming', 'is': 'is', 'a': 'a', 'very': 'very', 'creative': 'creative', 'and': 'a
nd', 'rewarding': 'rewarding', 'activity': 'activity', 'You': 'You', 'can': 'can
', 'write': 'write', 'for': 'for', 'many': 'many', 'reasons': 'reasons', 'rangin
g': 'ranging', 'from': 'from', 'making': 'making', 'your': 'your', 'living': 'li
ving', 'to': 'to', 'solving': 'solving', 'difficult': 'difficult', 'data': 'data
', 'analysis': 'analysis', 'problem': 'problem', 'having': 'having', 'fun': 'fun
', 'helping': 'helping', 'someone': 'someone', 'else': 'else', 'solve': 'solve',
'This': 'This', 'book': 'book', 'assumes': 'assumes', 'that': 'that', '{\\em':
'{\\em', 'everyone}': 'everyone}', 'needs': 'needs', 'know': 'know', 'how': 'how
', 'program': 'program', 'once': 'once', 'you': 'you', 'program,': 'program,',
will': 'will', 'figure': 'figure', 'out': 'out', 'what': 'what', 'want': 'want',
'do': 'do', 'with': 'with', 'newfound': 'newfound', 'skills': 'skills', 'We': '
We', 'are': 'are', 'surrounded': 'surrounded', 'in': 'in', 'our': 'our', 'daily'
: 'daily', 'lives': 'lives', 'computers': 'computers', 'laptops': 'laptops', 'ce
ll': 'cell', 'phones': 'phones', 'think': 'think', 'of': 'of', 'these': 'these',
'as': 'as', 'personal': 'personal', 'assistants': 'assistants', 'who': 'who', '
take': 'take', 'care': 'care', 'things': 'things', 'on': 'on', 'behalf': 'behalf
', 'The': 'The', 'hardware': 'hardware', 'current-day': 'current-day', 'essentia
lly': 'essentially', 'built': 'built', 'continuously': 'continuously', 'ask': 'a
sk', 'us': 'us', 'the': 'the', 'question': 'question', 'What': 'What', 'would':
'would', 'like': 'like', 'me': 'me', 'next': 'next', 'Our': 'Our', 'fast': 'fast
', 'have': 'have', 'vast': 'vast', 'amounts': 'amounts', 'memory': 'memory', '
could': 'could', 'be': 'be', 'helpful': 'helpful', 'if': 'if', 'we': 'we', 'only
': 'only', 'knew': 'knew', 'language': 'language', 'speak': 'speak', 'explain':
'explain', 'computer': 'computer', 'it': 'it', 'If': 'If', 'this': 'this', 'tell
': 'tell', 'tasks': 'tasks', 'were': 'were', 'reptitive': 'reptitive', 'Interest
ingly,': 'Interestingly,', 'kinds': 'kinds', 'best': 'best', 'often': 'often', '
humans': 'humans', 'find': 'find', 'boring': 'boring', 'mind-numbing': 'mind-num
bing'})
Introduce la llave a la que quieres acceder al valor: have
El valor es: have
```

Ejercicio 9.2 Escribe un programa que ordene en categorías cada mensaje de correo según el día de la semana en que fue hecho el envío. Para lograrlo, busca las líneas que comienzan por “From”, luego localiza la tercera palabra y mantén un contador actualizado de cada uno de los días de la semana. Al final del programa imprime en pantalla el contenido de tu diccionario (el orden no importa).

Línea de ejemplo: From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

Ejemplo de Ejecución: python dow.py

Introduzca un nombre de fichero: mbox-short.txt

{'Fri':20, 'Thu':6, 'Sat':1}

```
Ejercicio9.2.py x
1 file=open("mail.txt")
2 diccionario=dict()
3 cont=0
4 for linea in file:
5     cadena=linea.split()
6     if cadena[0]=='From:':
7         if cadena[2] not in diccionario:
8             diccionario[cadena[2]]=1
9         else:
10            diccionario[cadena[2]]+=1
11 print("Lista de correos:")
12 print(diccionario)
13
```

```
C:\Users\Alexandra\Downloads>python Ejercicio9.2.py
Lista de correos:
{'Lunes': 3, 'Martes': 4, 'Miercoles': 3, 'Viernes': 2, 'Jueves': 1}
```

Ejercicio 9.3 Escribe un programa que lea a través de un registro de correo, construya un histograma usando un diccionario para contar cuántos mensajes han llegado desde cada dirección de correo, e imprima el diccionario.

Introduzca un nombre del fichero: mbox-short.txt

```
{'gopal.ramasammycook@gmail.com':1, 'louis@media.berkeley.edu':3,
'cwen@iupui.edu':5, 'antranig@caret.cam.ac.uk':1, 'rjlowe@iupui.edu':2,
'gsilver@umich.edu':3, 'david.horwitz@uct.ac.za':4, 'wagnermr@iupui.edu':1,
'zqian@umich.edu':4, 'stephen.marquard@uct.ac.za':2, 'ray@media.berkeley.edu':1}
```

```
Ejercicio9.3.py x
1 file=open("mail.txt")
2 diccionario=dict()
3 cont=0
4 ▼ for linea in file:
5     cadena=linea.split()
6 ▼     if cadena[0]=='From:':
7         if cadena[1] not in diccionario:
8             diccionario[cadena[1]]=1
9         else:
10            diccionario[cadena[1]]+=1
11 print("Lista de correos:")
12 print(diccionario)
13
```

```
Símbolo del sistema

C:\Users\Alexandra\Downloads>python Ejercicio9.3.py
Lista de correos:
{'disasterology03@hotmail.com': 2, 'leugim98zurc@gmail.com': 3, 'aida
vela@hotmail.com': 2, 'omy@hotmail.com': 3, 'JoseL@hotmail.com': 1, '
Macz@sodvi.com': 1, 'kevmv@hotmail.com': 1}

C:\Users\Alexandra\Downloads>_
```

Ejercicio 9.4 Añade código al programa anterior para localizar quien tiene más mensajes en el fichero.

Después de que los datos hayan sido leídos y el diccionario haya sido creado, busca a través del diccionario usando un bucle máximo (mira la Section 5.7.2) para encontrar quien es el que tiene más mensajes e imprime cuántos mensajes tiene esa persona.

Introduzca un nombre de fichero: mbox-short.txt

cwen@iupui.edu 5

Introduzca un nombre de fichero: mbox.txt

zqian@umich.edu 195

```
Ejercicio9.4.py x
2  diccionario=dict()
3  cont=0
4  for linea in file:
5      cadena=linea.split()
6      if cadena[0]=='From:':
7          if cadena[1] not in diccionario:
8              diccionario[cadena[1]]=1
9          else:
10             diccionario[cadena[1]]+=1
11
12  mayor = None
13  for valor in diccionario:
14      if mayor is None or valor > mayor :
15          mayor = valor
16
17  print ("El correo: ",mayor,"tiene ",diccionario[mayor]
18        , "mensajes")
```

```
C:\> Símbolo del sistema

C:\Users\Alexandra\Downloads>python Ejercicio9.4.py
El correo: omy@hotmail.com tiene 3 mensajes

C:\Users\Alexandra\Downloads>_
```

Ejercicio 9.5 Este programa debe guardar el nombre de dominio desde donde se envió el mensaje en vez de quien mandó el mensaje (es decir, la dirección de correo completa). Al final del programa, imprime en pantalla el contenido de tu diccionario.

`python schoolcount.py`

Introduzca un nombre de fichero: mbox-short.txt

{'media.berkeley.edu':4, 'uct.ac.za':6, 'umich.edu':7, 'gmail.com':1, 'caret.cam.ac.uk':1, 'iupui.edu':8}

```
Ejercicio9.5.py x
1 file=open("mail.txt")
2 diccionario=dict()
3 cont=0
4 for linea in file:
5     cadena=linea.split()
6     if cadena[0]=='From:':
7         dominio=cadena[1].split("@")
8         if dominio[1] not in diccionario:
9             diccionario[dominio[1]]=1
10        else:
11            diccionario[dominio[1]]+=1
12 print("Lista de correos:")
13 print(diccionario)
14
```

```
C:\> Símbolo del sistema

C:\Users\Alexandra\Downloads>python Ejercicio9.5.py
Lista de correos:
{'hotmail.com': 9, 'gmail.com': 3, 'sodvi.com': 1}

C:\Users\Alexandra\Downloads>_
```