Gustavo Vargas Ramírez
200409141

CAJA DE ANDE · CARBONO NEUTRAL+

Tree diagram with nodes: 50, 17, 76, 9, 23, 54, 14, 19, 72, 12, 67 — with "Act" and "Null" labels on branches.
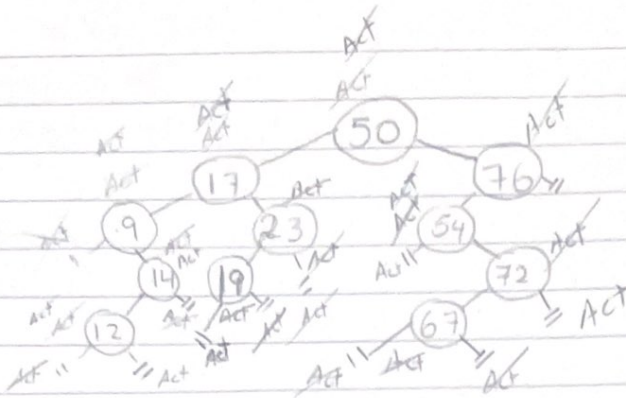
Iterativos

Preorden → 50 — 17 — 9 — 14 — 12 — 23 — 19 — 76 — 54 — 72 — 67

Preorden I

NB "Act = 50

p.vacia? = sí
while
    Act != NULL sí
cout 50
    p.Push (Act)

Act = Act → Hizq

Act != NULL? sí
cout 17
p.Push (Act)

Act = Act → Hizq

Act != NULL sí
cout 9
p.Push (Act)

Act = Act → Hizq

p.vacia() ==false? sí

Act != NULL? NO
else
    Act.Pop()

Act = Act → Hder

Act != NULL? sí
cout 14
p.push (Act)

Act = Act → Hizq

Act != NULL? sí
cout 12
p.push (Act)

Act = Act → Hizq

p.Vacia = false? sí
Act != NULL NO
Act = p.Pop()
Act = Act → Hder

P
67
72
54
76
19
23
12
14
9
17
50

p.Vacia = false? Sí
Act != NULL? NO
Act = p.Pop

Act = Act → Hder;

p.Vacia = false? sí
Act != NULL? NO

Act = p.Pop

Act = Act → Hder

Act != NULL? Sí
cout 23
p.push(Act)

Act = Act → Hizq

Act != Nullo? Sí
cout 19
p.push (Act)

Act = Act → Hder

p.Vacia = false? sí
Act != NULL NO
Act = Act.pop()

Act = Act → Hder()

---

p.Vacia = false? Sí
Act != NULL? NO
Act = Act pop()

Act = Act → Hder

p.vacia = false? sí
Act != NULL? NO

Act = p.pop() 50
Act = Act → Hder

Act != NULL? Sí
cout 76
p.push (Act)

Act = Act → Hizq

Act != NULL? Sí
cout 54
p.push (Act)

Act = Act → Hizq

p.Vacia = false? sí

Act != NULL? NO

Act = p.pop()

Act = Act → Hder()

Act != NULL? Sí
cout 72
p.push (Act)

Act = Act → Hizq

Act =! NULL? Sí
cout 67

p.push (Act)

Act = Act → Hizq

---

p.vacia = false? sí
Act != NULL NO
Act.pop()

Act = Act → Hder
p.vacia = false? sí
Act != NULL? NO
Act pop()
Act = Act → Hder

p.vacia = false? sí

Act != NULL NO

Act.pop()

Act = Act → Hder

p.vaca = false? NO

Act != NULL? NO

END

notas

if(p.Vacia() & Act==NULL){

Iterativo

Inorden

NB `Act = 50

band = true
while (band)
  while (Act != NULL)
    p.push(50)
    Act = Act→Hizq
    P.push (17)
    Act = Act→Hizq
    P.push (9)
    Act = Act→Hizq
  !p.Vacia() ? sí
  Act = p.Pop() (9)
  cout 9
  Act = Act→Hder
  while(Act != NULL)
    P.push(14)
    Act = Act→Hizq
    p.push (12)
    Act = Act→Hizq
  !p.Vacia() ? yes
  Act = p.Pop() (12)
  cout 12
  Act = Act→Hder

  Act != NULL? NO
  !p.Vacia() yes
  Act = p.pop() (14)
  cout 14
  Act = Act→Hder

  Act != NULL ? NO
  !p.vacie? sí
  Act = p.pop() (17)
  cout 17
  Act = Act→Hder

  while(Act != NULL)
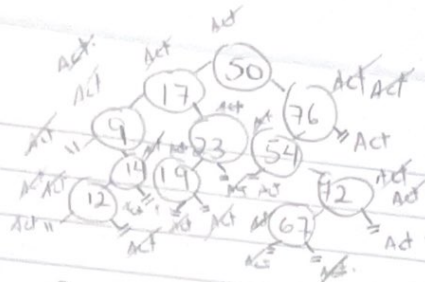   P.push(23)
    Act = Act→Hizq
    P.push(19)
    Act = Act→Hizq

67	
72	
59	
76	
19	
23	
12	
14	
9	
17	
50	

p 50

Act != NULL NO
if !p.Vacia ? sí
Act = p.Pop() (19)
cout 19
Act = Act→Hder

Act != NULL NO
if !p.Vacia ? sí
Act = p.Pop()(23)
Cout 23
Act = Act→Hder

Act != NULL NO
if !p.Vacia)? yes
Act = p.pop() (50)
cout 50
Act = Act→Hder

Act != NULL ? sí
P.push (76)
Act = Act→Hizq

Act != NULL? sí
p.push(54)
Act = Act→Hizq

Act != NULL? NO
!p.vacia)? sí
Act = p.pop() 54
cout 54
Act = Act→Hder

Act != NULL ? sí
p.push (72)
Act = Act→Hizq

Act != NULL? sí
P.push(67)
Act = Act→Hizq

Act != NULL? NO
!p.vacia()? sí
Act = p.pop() (67)
cout 67
Act = Act→Hder

Act != NULL? NO
!p.vacia? sí
Act = p.pop()(72)
cout 72
Act = Act→Hder

9 - 12 - 14 - 17 - 19 - 23 - 50 - 54 - 67 - 72 - 76 -

Act != NULL ? NO
!p.vacia)? sí
Act = p.pop()(76)
cout 76
Act = Act→Hder

Act != NULL? NO
!p.vacia ? NO

p.vacia & Act == NULL? sí

Break.

Iterativo
   Post Orden

NB *Act = 50
while (!p.vacia || Act != NULL)

   Act != NULL? sí

   p2.push (new NB(50))

   p.push (50)

   Act = Act→Hder

   Act != NULL? sí

   p2.push(new NB(76))

   p.push (76)

   Act = Act→Hder

else Act= p.pop() (76)

   Act = Act→Hizq

   Act != NULL? sí

   p2.push (new NB(54))

   p.push(54)

   Act = Act→Hder

   Act != NULL? sí

   p2.push (new NB(72))

   p.push(72)

   Act = Act→Hder

else Act = p.pop() (72)

   Act = Act→Hizq

   Act != NULL? sí

   p2.push (new NB(67))

   p.push (67)
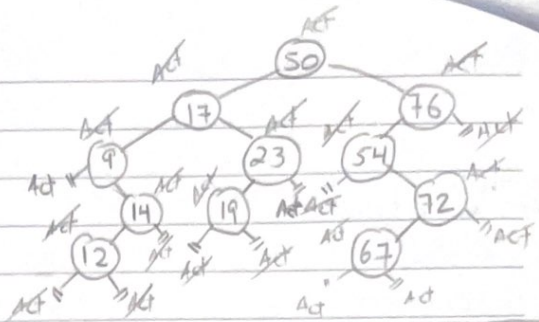
   Act = Act→Hder

else Act= p.pop() (67)

   Act = Act→Hizq

else Act = p.pop() (54)

   Act = Act→Hizq

---

else Act = p.pop() (50)

   Act = Act→Hizq

   Act != null? sí

   p2.push (New NB(17))

   p.push (17)

   Act = Act→Hder

   Act != null? sí

   p2.push (new NB(23))

   p.push (23)

   Act = Act→Hder

else Act = p.pop() (23)

   Act = Act→Hizq

   Act != null? sí

   p2.push (new NB(19))

   p.push (19)

   Act = Act→Hder

else Act = p.pop() (19)

   Act = Act→Hizq

else   Act = p.pop() (17)

   Act = Act→Hizq

   Act != null? sí

   p2.push (new NB(9))

   p.push (9)

   Act = Act→Hder

   Act != null? sí

   p2.push (new NB(14))

   p.push (14)

   Act = Act→Hder

---

else Act = p.pop() (14)

   Act = Act→Hizq

   Act != null? sí

   p2.push (new NB(12))

   p.push (12)

   Act = Act→Hder

else Act = p.pop() (12)

   Act = Act→Hizq

   Act = p.pop() (9)

   Act = Act→Hizq

exit while #1

while #2

W! p2.vacia() ? sí

NB *salida = p2.pop() (12)
cout 12
NB *salida = p2.pop() (14)
cout 14
NB *salida = p2.pop() (9)
cout 9
NB *salida = p2.pop() (19)
cout 19
NB *salida = p2.pop() (23)
cout 23
NB *salida = p2.pop() (17)
cout 17
NB *salida = p2.pop() (67)
cout 67
NB *salida = p2.pop() (72)
cout 72
NB *salida = p2.pop() (54)
cout 54
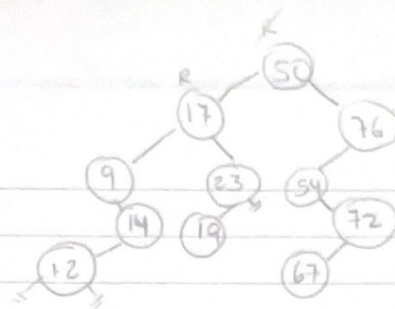NB *salida = p2.pop() (76)
cout 76
NB *salida = p2.pop() (50)
cout 50

!p2.vacia() ? NO

   exit while #2

12-14-9-19-23-17-67-72-54-76-50

Recursivo / Preorden

R=50
if
  R = null ? NO
  cout 50
else
  preordenR(17)
  preordenR(76)

  → R = NULL? NO
  cout 17
  preorden R(9)
  preorden R(23)

  → R = NULL? NO
  cout 9
  PreordenR(NULL)
  preordenR(14)

  → R = Nll? sí
  return

  → R = NULL? NO
  preorden R(12)
  preorden R(null)

  → R = null? NO
  cout 12
  preorden R(null)
  preorden R(null)

  → return
  → return
  → return

  → R = null? NO
  cout 23
  PreordenR(19)
  preorden R(null)

  → R = null? no
  cout 19
  preorden R(null)
  preorden R(null)

  → return
  → return
  → return

R = null? NO
cout 76
preorden R(54)
preorden R(null)

→ R = null? NO
cout 54
preordenR(null)
preorden R(72)

→ return
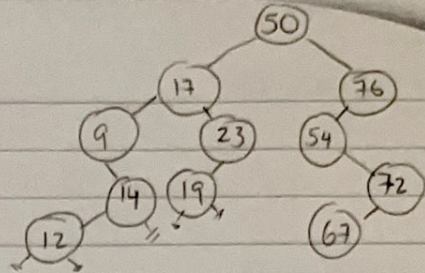
→ R = null? NO
cout 72
preorden R(67)
preorden R(null)

→ R = null? NO
cout 67

→ Return

→ Return

50 - 17 - 9 - 14 - 12 - 23 - 19 - 76 - 54 - 72 - 67

Recursivo / InOrden R
R=50
R=null? NO

InOrden (17)
R=null? NO

Inorden (9)
R=null? NO

Inorden (null) (9 Hizq)
return

cout 9

Inorden (14)
R=null? NO

Inorden (12)
R=null? NO

inorden (null) (12 Hizq)
return

cout 12

inorden (null) (12 Hder)
return

inorden (null) (14 Hder)
return

cout 14

inorden (null) 14 Hder
return
cout 17
inorden (23)
r= null? NO

inorden (19)
r= null? no

inorden (null) (19 Hizq
return

cout 19

inorden (null) (19 Hder)
return

cout 23

inorden (null) (23 Hder)
return

cout 50

Inorden (76)   (50 Hder)

R=null? NO.

Inorden(54)

Inorden (null) (54 Hizq)
return

cout 54

Inorden (72)

Inorden 67
Inorden (null) (67 Hizq)
return
cout 67

Inorden (null) (67 Hder)
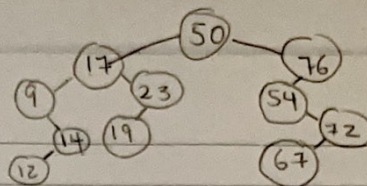return

Cout 72

Inorden (null) (72 Hder)
return

cout 76

Inorden (NULL)   (76 Hder)

return;

---

9 - 12 - 14 - 17 - 19 - 23 - 50 - 54 - 67 - 72 - 76

Tree: 50 → 17, 76; 17 → 9, 23; 76 → 54; 9 → 14; 23 → 19; 54 → 72; 9 → 12; 72 → 67

Recursivo / PostOrden
R=50
Postorden (50)

R = NULL?   NO
else

12 - 14 - 9 - 19 - 23 = 17 - 67 - 72 - 54 - 76 - 50

—— postorden (17)

  R=null? NO
  else
—— —— postorden (9)

    R=null? NO
    else
—— —— —— postorden (null) (17 Hizq)
          return

—— —— postorden (14)

—— —— —— postorden (12)

—— —— —— —— postorden (NULL) (12 Hizq)
              return
                postorden (null) (12 Hder)
              return
—— —— —— cout 12

—— —— —— postorden (null) (14 Hder)
              return
—— —— cout 14

      cout 9

—— —— postorden (23)

—— —— —— postorden (19)

—— —— —— —— postorden (null) (19 Hizq)
              return
                postorden (null) (19 Hder)
—— —— —— cout 19

—— —— —— postorden (null) (23 Hder)
              return
—— —— cout 23

—— cout 17

— postorden (76)

—— —— postorden (54)

—— —— —— postorden (null)   (54 Hizq)

—— —— —— postorden (72)

—— —— —— —— postorden (67)
—— —— —— —— —— postorden (null) (67 Hizq)
                postorden (null)(67 Hder)
—— —— —— cout 67

Right column:

—— —— —— postorden (null) (72 Hder)

—— —— — cout 72

—— cout 54

—, postorden (null) (76 Hder)

— cout 76

cout 50

    END

valor));