

Programación Científica y HPCI

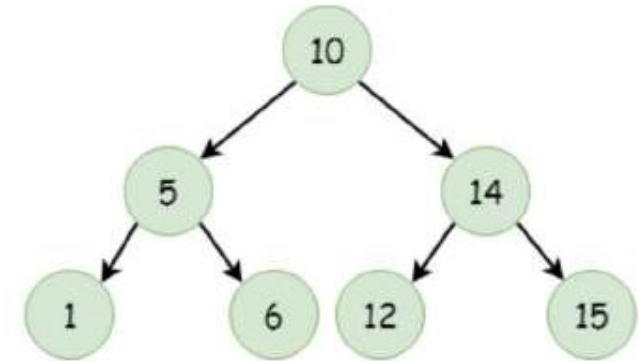
Máster Universitario en Ingeniería Matemática y Computación

Actividad 1 – Directrices de ayuda

Tarea 1. Recorrido del árbol en profundidad de forma iterativa

1. Recorrido en pre-orden
2. Recorrido en in-orden

Recorrido en pre-orden



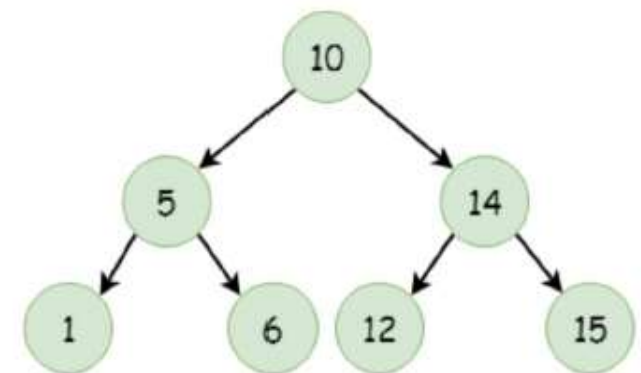
[10, 5, 1, 6, 14, 12, 15]

1. Se apila el nodo raíz
2. Mientras la pila no esté vacía
 - a. Desapilar y guardar en una lista
 - b. Guardar hijo derecho en la pila.
 - c. Guardar hijo izquierdo en la pila.

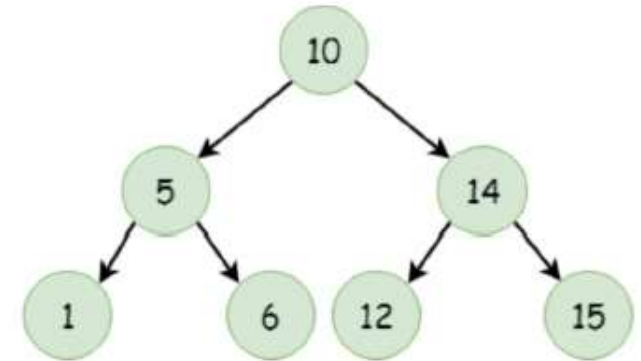
Recorrido en pre-orden

Pila	Lista
10	[]
5 14	[10]
1 6 14	[10,5]
6 14	[10,5,1]
14	[10,5,1,6]
12 15	[10,5,1,6,14]
15	[10,5,1,6,14,12]
	[10,5,1,6,14,12,15]

[10, 5, 1, 6, 14, 12, 15]



Recorrido en in-orden



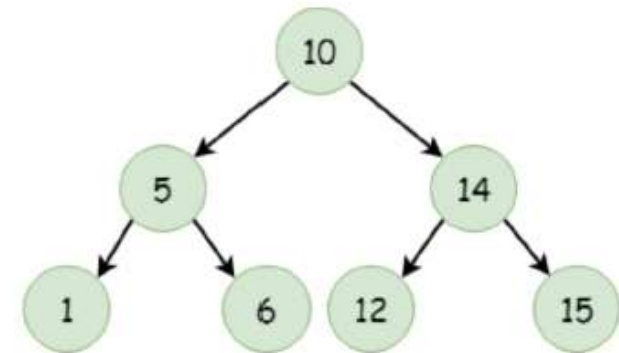
[1, 5, 6, 10, 12, 14, 15]

1. Obtener la raíz y asignarlo a una variable actual .
2. Si actual no es NULL
 - a. Apilar actual en la pila.
 - b. Asignar a actual al hijo izquierdo de actual y pasar al paso 2.
3. Si actual es NULL y la pila no está vacía
 - a. sacar el elemento de la pila y meterlo en la lista
 - b. Guardar en actual al hijo derecho del que sea acaba de obtener
 - c. Ir al paso 2.

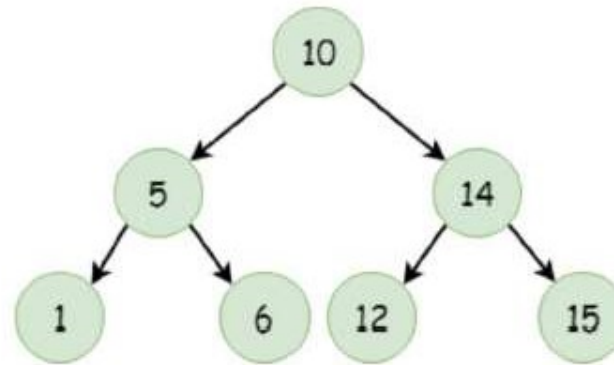
Actual	Pila	Lista
10		
5	10	[]
1	5 10	[]
null	1 5 10	[]
null	5 10	[1]
6	10	[1,5]
null	6 10	[1,5]
null	10	[1,5,6]
14		[1,5,6,10]
12	14	[1,5,6,10]
null	12 14	[1,5,6,10]
null	14	[1,5,6,10,12]
15		[1,5,6,10,12,14]
null	15	[1,5,6,10,12,14]
null		[1,5,6,10,12,14,15]

Recorrido en in-orden

[1, 5, 6, 10, 12, 14, 15]



Tarea 2. Recorrido en amplitud



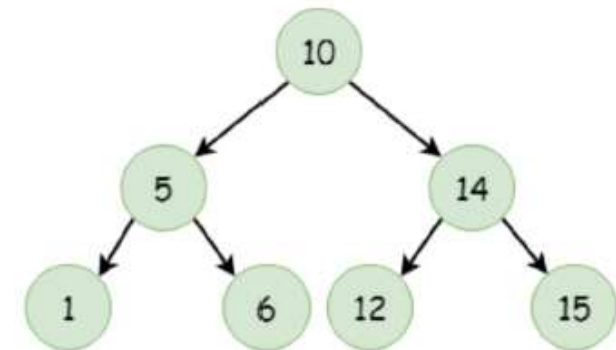
[10, 5, 14, 1, 6, 12, 15]

- ▶ Se visitan los nodos por niveles
- 1. Se introduce la raíz en la cola
- 2. Se repite:
 - a. Quitar primer elemento de la cola,
 - b. Guardar en la lista el contenido de dicho nodo
 - c. Almacenar en la cola sus hijos izquierdo y derecho.

Recorrido en amplitud

Cola	Lista
10	[]
5, 14	[10]
14 , 1, 6	[10,5]
1,6,12,15	[10,5,14]
6,12,15	[10,5,14,1]
12,15	[10,5,14,1,6]
15	[10,5,14,1,6,12]
	[10,5,14,1,6,12,15]

[10, 5, 1, 6, 14, 12, 15]



Rendimiento de una función

```
import profile
from time import time

def ordenacionBurbuja(lista):
    'Variable global que puede se usada fuera de la función'
    global numComparaciones
    n = len(lista)

    for i in range(1, n):
        for j in range(n-i):
            numComparaciones += 1

            if lista[j] > lista[j+1]:
                'intercambio de valores'
                lista[j], lista[j+1] = lista[j+1], lista[j]

lista = [35, 16, 17, 73, 8, 0]
numComparaciones = 0

t0 = time()
ordenacionBurbuja(lista)
t1 = time()

print ("Lista ordenada:")
print(lista)

print ("Tiempo: {0:f} segundos".format(t1 - t0))
print ("Comparaciones:", numComparaciones)

profile.run('print(ordenacionBurbuja(lista))')
```

ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
18454895/33	4.665	0.000	4.665	0.141	<stdin>:1(fib)
33/1	0.000	0.000	4.665	4.665	<stdin>:1(fib_seq)
1	0.000	0.000	4.666	4.666	<string>:1(<module>)
1	0.000	0.000	4.666	4.666	{built-in method builtins.exec}
1	0.000	0.000	0.000	0.000	{built-in method builtins.print}
33	0.000	0.000	0.000	0.000	{method 'append' of 'list' objects}
1	0.000	0.000	0.000	0.000	{method 'disable' of '_lsprof.Profiler' obj}
32	0.000	0.000	0.000	0.000	{method 'extend' of 'list' objects}



www.unir.net