**SEGUNDO PARCIAL**

**INF310 SX**‒ **Estructuras de Datos II. Gestión 1-2018.**

**Subgrupo: Apellidos A-L**

Arbol M-Vías

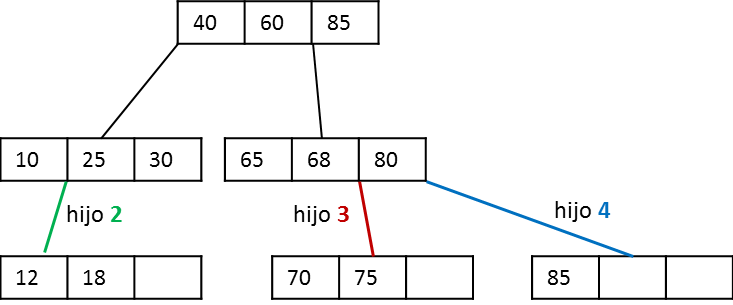
1. En la class ArbolM, escriba la función:

public int sum()

la cual calcule la *suma ponderada* de las **hojas**, la cual se calcula así:

* Si el árbol es **vacío** o es una **hoja**, sum=**0**.
* Sea s(h) la suma de los Data’s de la hoja h multiplicada por el **número de hijo** que la sostiene. Si el árbol tiene las hojas h1, h2, …, hn, entonces, sum=s(h1) + s(h2) + … +s(hn)

**/\*** *Entendiendo el concepto*



hoja h1 hoja h2 hoja h3

s(h1) = suma de las datas de la hoja h1 x el número de hijo que la sostiene = (12 +18) x **2** = 60

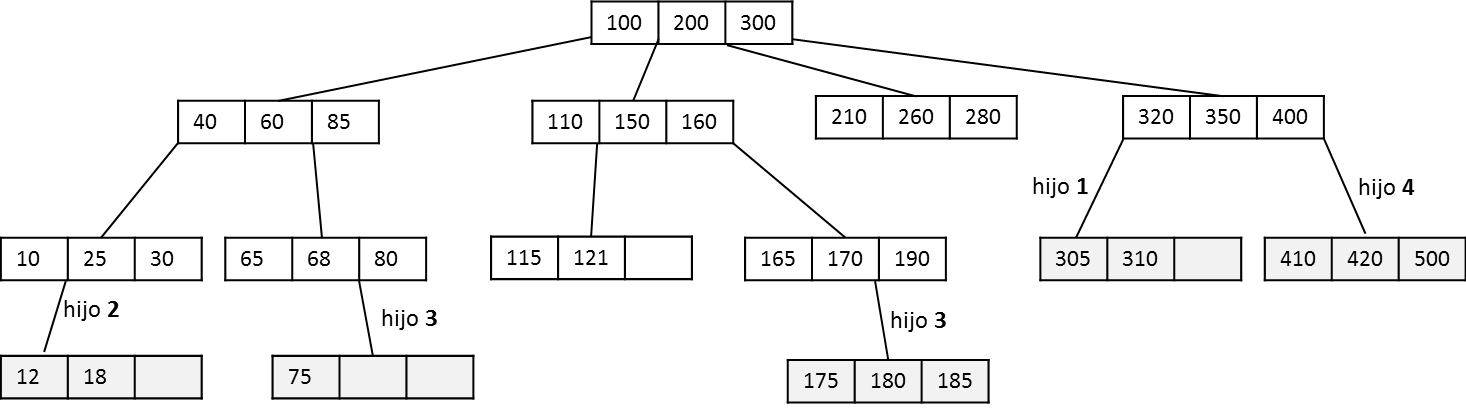
s(h2) = suma de las datas de la hoja h2 x el número de hijo que la sostiene = (70 +75) x **3** = 290

s(h3) = suma de las datas de la hoja h3 x el número de hijo que la sostiene = (85) x **4** = 340

Entonces la sum de éste árbol es: sum( ) = s(h1) + s(h2) + s(h3) = 60 + 290 + 340 = 690

**\*/**

Por ejemplo: Si tomamos el siguiente árbol A:



**A.sum()** = (12+18)\***2** + (75)\***3** + (175+180+185)\***3** + (305+310)\***1** + (410+420+500)\***4** = 60 + 225 + 1620 + 615 + 5320 = **7840**

*Tome en cuenta que se puede saber cuántas Datas usadas tiene un NodoM N, usando la función N.cantDatasUsadas().*

Grafos Pesados (TagGraph)

2. En la class Grafo, escriba la función

public boolean isAlcanzable(int a, int z, int msj)

la cual devuelva true si y solo si:

* Si el **msj** es **par**, partiendo de **a** se alcanza **z**, usando solo aristas con **peso par**.
* Si el **msj** es **impar**, partiendo de **a** se alcanza **z**, usando solo aristas con **peso impar**.

*Tome en cuenta que la función costo(u, v), implementada en la class Grafo, devuelve cero si* ***no*** *existe la arista u → v.*

Por ejemplo:

**35**

**50**

**25**

**30**

**15**

**10**

**20**

**10**

**25**

**25**

**10**

**12**

**5**

**5**

isAlcanzable(0, 1, 25)=true //Como 25 es impar, partiendo del 0 puede llegar hasta el 1 usando solo aristas con pesos impares: 0→3→1

isAlcanzable(4, 2, 80)=false //Como 80 es par, partiendo del 4 no se puede llegar hasta el 2 usando solo aristas con pesos pares.

isAlcanzable(7, 8, 20)=false

isAlcanzable(8, 7, 20)=true //Como 20 es par, partiendo del 8 se puede llegar hasta el 7 usando solo aristas con pesos pares: 8→7

isAlcanzable(0, 6, 11)=false //Como 11 es impar, partiendo del 0 no se puede llegar hasta el 6 usando solo aristas con pesos impares.

isAlcanzable(5, 2, 40)=true //Como 40 es par, partiendo del 5 se puede llegar hasta el 2 usando solo aristas con pesos pares: 5→3→6→2