

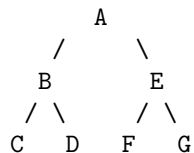
Exercici 2. Escriu una funció **recursiva** de nom *nestedListSum*, tal que, donada una llista de llistes niuades, retorni el resultat de calcular recursivament la suma d'elements inclosos en la llista/subllistes.

```
def nestedListSum(NL):
    """
    >>> nestedListSum([2,4,8])
    14
    >>> nestedListSum([[1,4],[2,8],[3,1,1]])
    20
    >>> nestedListSum([[2],[1]],[2,4],[[3,4],[1,0,-1]])
    16
    """
```

Exercici 3. Donada la definició de la classe arbre binari treballada a classe de teoria, es demana:

Apartat a) implementeu un nou mètode recursiu, *recorregutInordre*, que escrigui per pantalla els nodes de l'arbre en inordre. Un recorregut en inordre d'un arbre consisteix en, per cada node, realitzar un recorregut en inordre de l'arbre esquerre, visitar el node, i després realitzar un recorregut en inordre de l'arbre dret.

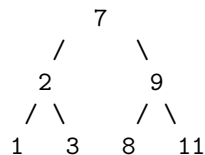
Per exemple, donat l'arbre,



el resultat del recorregut en inordre hauria de ser CBDAFEG

Apartat b) implementeu un nou mètode recursiu, *comptaNodesParells*, tal que, donat un arbre binari de nombres, compti quants dels nodes són parells.

Per exemple, donat l'arbre,



El mètode hauria de retornar el valor 2.

Exercici 4. Anàlisi de la complexitat

Donada una llista de n elements, respon les preguntes següents.

1. El cost asimptòtic en cas pitjor de l'algorisme d'ordenació **Merge Sort** és:
2. El cost asimptòtic en cas pitjor de l'algorisme d'ordenació **Bubble Sort** és:
3. El cost asimptòtic en cas pitjor de l'algorisme d'ordenació per **selecció** és:
4. Justifica amb un exemple el funcionament de l'algorisme d'**ordenació per selecció** i el seu cost.
5. Donada una **queue**, el cost del mètode *put* és ... i el cost del mètode *remove* és ...
6. Donada una **stack**, el cost del mètode *push* és ... i el cost del mètode *pop* és ...