

---

## L'Arbre Escorat

T36888\_ca

---

Donat un arbre binari  $a$ , amb nombres enters positius com a valor dels nodes, volem obtenir el seu equivalent *escorat*. Veiem què significa que un arbre estigui *escorat*.

Comencem definint el *pes* i la *mida* d'un arbre binari: Els nombres en els nodes representen pesos. El *pes* d'un arbre és la suma dels valors en els seus nodes. La *mida* d'un arbre és el nombre de nodes que té.

Volem *escorar* l'arbre de tal manera que, per a tot node, el fill esquerre sempre pesi menys que el fill dret. Així, si el pes del fill esquerre és superior al pes del fill dret, l'arbre escorat serà l'arbre amb la mateixa arrel, el fill dret com a fill esquerre, i el fill esquerre com a fill dret (s'intercanvien de costat els fills). En cas que els dos fills pesin el mateix, els intercanviarem si la mida del fill esquerre és més gran que la mida del fill dret.

Noteu que a l'arbre hi haurà els mateixos nodes abans i després d'escorar, però col·locats de manera diferent, i també hi ha els mateixos camins abans i després però disposats de manera diferent.

Us demanem, doncs, que feu una funció **escorar(a)** i l'afegiu al fitxer **code.py** tal com s'especifica a les Observacions.

### Paràmetres i retorn

El paràmetre de la funció **escorar(a)** és una instància de la classe `ArbreBinari` (que ja coneixem).

Els valor en cada node de l'arbre és un nombre enter positiu.

La funció **escorar(a)** ha de retornar *tres coses*: l'arbre escorat (instància d'`ArbreBinari`), el pes de l'arbre (un nombre enter) i la mida de l'arbre (un nombre enter).

### Entrada

L'entrada al programa serà el preordre de l'arbre binari, amb marca -1 per indicar els arbres buits (ja coneixem aquest format dels exercicis a classe de laboratori).

Vegeu els exemples del joc de proves públic.

### Sortida

El programa ha d'escriure el preordre i l'inordre de l'arbre escorat, amb els valors dels nodes separat per ',', sense marques per representar els arbres buits. El preordre i l'inordre han d'estar en línies separades

Vegeu els exemples del joc de proves públic.

### Observacions

Heu de baixar-vos el fitxer **code.py** (icona de la serp). Aquest fitxer és un programa amb **tot** el que cal per executar els jocs de prova públics. Només falta, clar, la funció que us demana l'enunciat. Aquest fitxer l'heu de completar amb el codi que falta, i això, **tot**, és el que heu d'enviar al Jutge com a solució.

Dins el fitxer **code.py** teniu la classe **ArbreBinari** que hem treballat a les classes de teoria. Hem tret alguns mètodes que segur no tenen cap utilitat per resoldre aquest problema. No

caldrà que la vostra solució faci cap *import* ni res. Tot el codi que us cal el teniu dins de `code.py`.

L'eficiència i la qualitat de la solució es tindran en compte a la correcció manual.

#### Exemple d'entrada 1

```
11 2 3 -1 -1 10 -1 -1 3 5 -1 -1 4 -1 -1
```

#### Exemple de sortida 1

```
11,3,4,5,2,3,10
4,3,5,11,3,2,10
```

#### Exemple d'entrada 2

```
5 2 10 -1 -1 8 -1 -1 20 -1 -1
```

#### Exemple de sortida 2

```
5,20,2,8,10
20,5,8,2,10
```

#### Exemple d'entrada 3

```
5 2 -1 -1 -1
```

#### Exemple de sortida 3

```
5,2
5,2
```

#### Exemple d'entrada 4

```
10 2 3 -1 -1 5 -1 -1 2 3 -1 -1 5 -1 -1
```

#### Exemple de sortida 4

```
10,2,3,5,2,3,5
3,2,5,10,3,2,5
```

#### Exemple d'entrada 5

```
10 -1 -1
```

#### Exemple de sortida 5

```
10
10
```

### Informació del problema

Autor : Jordi Delgado (amb agraïments als professors de PRO2-GEI)

Generació : 2025-04-02 10:49:02

© *Jutge.org*, 2006–2025.

<https://jutge.org>