
L'Arbre Escorat (2)**U15608_ca**

Donat un arbre (general) de naturals, el seu *pes* és la suma dels valors en els seus nodes i la seva *mida* és el nombre de nodes que té.

Escorar un arbre és transformar-lo de manera que, per a tots els seus nodes, els fills estiguin ordenats creixentment per pes, d'esquerra a dreta. En cas que dos fills pesin el mateix, els ordenarem d'acord a la seva mida, també creixentment.

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 **Arbre**

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

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

Entrada

L'entrada al programa serà la descripció d'un arbre en preordre, en el que al valor de cada node li segueix el seu nombre de fills.

Vegeu els exemples del joc de proves públic.

Sortida

El programa ha d'escriure el preordre i el postordre de l'arbre escorat, amb els valors dels nodes separat per ','. El preordre i el postordre 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 **Arbre**. Enteneu-la abans de procedir. És molt senzilla. No caldrà que la vostra solució faci cap *import* ni *res*. Tot el codi que us cal el teniu dins de **code.py**.

Cal considerar que la classe **Arbre** no té definit l'operador **<** per tant no es poden comparar les instàncies d'**Arbre**. Si feu servir **sort**, recordeu que hi ha un paràmetre **key** que us permet extreure la informació que interessa comparar d'allò que esteu ordenant.

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

Exemple d'entrada 1

```
7 3 8 0 4 2 3 1 0 1 6 0
5 0 2 4 9 0 1 0 8 0 5 0
```

Exemple d'entrada 2

```
2 0
```

Exemple d'entrada 3

```
1 3 2 2 5 0 6 0 3 1 7 2
11 0 12 0 4 3 8 0 9 0 10 0
```

Exemple d'entrada 4

```
1 2 100 2 50 3 15 0 10 0
5 0 30 0 10 2 3 0 2 1 1 0
```

Exemple de sortida 1

```
7, 8, 4, 5, 3, 0, 6, 2, 1, 5, 8, 9
8, 5, 6, 0, 3, 4, 1, 5, 8, 9, 2, 7
```

Exemple de sortida 2

```
2
2
```

Exemple de sortida 3

```
1, 2, 5, 6, 4, 8, 9, 10, 3, 7, 11, 12
5, 6, 2, 8, 9, 10, 4, 11, 12, 7, 3, 1
```

Exemple de sortida 4

```
1, 10, 3, 2, 1, 100, 30, 50, 5, 10, 15
3, 1, 2, 10, 30, 5, 10, 15, 50, 100, 1
```

Informació del problema

Autoria: Jordi Delgado

Generació: 2026-01-25T12:58:22.251Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>