

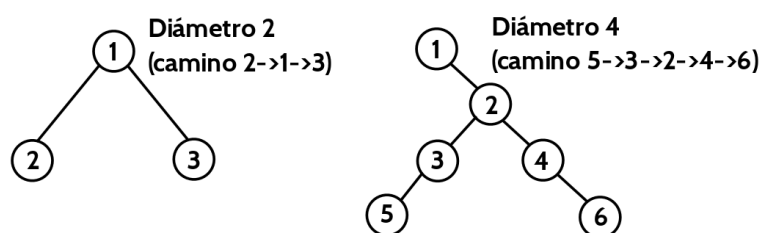
## Diámetro del árbol

S50568\_es

Dado un árbol binario **a**, queremos obtener su *diámetro*.

El *diámetro* de un árbol binario se define como el número de aristas del camino *más largo* entre dos nodos cualesquiera. No puede haber ni nodos repetidos ni aristas repetidas en ese camino. El diámetro del árbol vacío es 0. Es irrelevante el contenido (el valor) de los nodos, esta propiedad es una propiedad exclusivamente de la estructura del árbol. Fijaos que *el camino en cuestión no tiene por qué pasar por la raíz del árbol*.

Por ejemplo:



Os pedimos, pues, que hagáis una función **diametre(a)** y la añadáis al archivo **code.py** tal y como se especifica en las Observaciones.

### Parámetros y retorno de la función pedida

El parámetro de la función **diametre(a)** es una instancia de la clase **ArbreBinari** (que ya conocemos).

La función **diametre(a)** ha de retornar *dos cosas*: el diámetro del árbol (un número entero) y la altura del árbol (un número entero; número de nodos entre la raíz y alguna de las hojas más profundas, ambos inclusive). Recordad que tanto el diámetro como la altura del árbol vacío son 0.

### Entrada

La entrada del programa será el preorden del árbol binario, con marca -1 para indicar los árboles vacíos (ya conocemos este formato de los ejercicios en clase de laboratorio). Fijémonos en que no nos importa qué contienen los nodos del árbol.

Ved los ejemplos del juego de pruebas público.

### Salida

El programa ha de escribir el diámetro del árbol leído.

Ved los ejemplos del juego de pruebas público.

### Observaciones

Debe descargar el archivo **code.py** (icono de la serpiente). Este archivo es un programa con **todo** lo necesario para ejecutar los juegos de prueba públicos. Sólo falta, claro, la función que se pide en el enunciado. Este archivo debe completarse con el código que falta, y eso, **todo**, es lo que ha de enviarse al Jutge como solución.

Dentro del archivo `code.py` tenéis la clase `ArbreBinari` que hemos trabajado en las clases de teoría y laboratorio. Hemos quitado algunos métodos que seguro no tienen ninguna utilidad para resolver este problema. No hará falta que vuestra solución haga ningún *import* ni nada. Todo el código que necesitáis lo tenéis dentro de `code.py`.

Os hacemos retornar tanto el diámetro del árbol como la altura porque esta es *toda* la información que necesitáis *de los hijos* para conocer el diámetro y la altura del árbol.

La eficiencia y la calidad de la solución se tendrán en cuenta en la corrección manual.

#### Ejemplo de entrada 1

```
0 0 0 -1 -1 0 -1 -1 0 0 -1 -1 0 -1 -1
```

#### Ejemplo de salida 1

```
4
```

#### Ejemplo de entrada 2

```
0 0 0 -1 -1 0 -1 -1 0 -1 -1
```

#### Ejemplo de salida 2

```
3
```

#### Ejemplo de entrada 3

```
0 0 -1 -1 -1
```

#### Ejemplo de salida 3

```
1
```

#### Ejemplo de entrada 4

```
0 0 0 -1 -1 0 -1 -1 0 0 -1 -1 0 -1 -1
```

#### Ejemplo de salida 4

```
4
```

#### Ejemplo de entrada 5

```
0 -1 -1
```

#### Ejemplo de salida 5

```
0
```

#### Ejemplo de entrada 6

```
0 0 -1 -1 0 -1 -1
```

#### Ejemplo de salida 6

```
2
```

#### Ejemplo de entrada 7

```
0 -1 0 0 0 -1 -1 -1 0 -1 0 -1 -1
```

#### Ejemplo de salida 7

```
4
```

### Información del problema

Autoría: Jordi Delgado

Generación: 2026-04-01T05:09:36.696Z

© Jutge.org, 2006–2026.

<https://jutge.org>