


```
int area_mayor_sin_barreras(const BinTree<bool> &tree)
```

Esta función debe devolver el tamaño del área más grande del árbol de booleanos pasado como parámetro, suponiendo que los nodos barrera son aquellos que tienen el valor `true`. Por ejemplo, para el árbol mostrado en la figura anterior, la función debe devolver 4.

2. Indicar el coste, en el caso peor, de la función anterior. El coste debe estar expresado en función del número de nodos del árbol de entrada. Indica también la recurrencia utilizada en el caso de llamadas recursivas.

Entrada

La entrada comienza con un número que indica el número de casos de prueba que vienen a continuación. Cada caso de prueba consiste en una línea con la descripción de un árbol binario mediante la notación vista en clase. El árbol vacío se representa mediante `.` y el árbol no vacío mediante `(iz x dr)`, siendo `x` la raíz, y `iz`, `dr` las representaciones de ambos hijos. Con respecto a los valores de los nodos, el valor `0` significa *false*, y el valor `1` significa *true*. El tamaño de cada árbol está comprendido entre 0 y 20000.

Salida

Para cada árbol se escribirá una línea con el valor devuelto por la función `area_mayor_sin_barreras`.

Entrada de ejemplo

```
4
(((( ( 0 . ) 0 ( . 0 . ) ) 0 (( . 1 . ) 1 . ) ) 1 . ) 0 ((( ( 0 . ) 0 ( . 0 . ) ) 1 ( . 0 . )))
((( ( 0 . ) 0 ( . 0 . ) ) 0 (( . 0 . ) 0 . ))
(( . 0 . ) 1 ( . 0 . ))
.
```

Salida de ejemplo

```
4
6
1
0
```

Créditos

Manuel Montenegro