
Suma máxima de un camino entre dos nodos cualquiera del árbol binario X61481_es

Implementa una función **RECURSIVA** que, dado un árbol binario de números enteros, obtén la suma máxima de un camino entre dos nodos cualquiera del árbol binario. Este sería la cabecera:

```
// Pre: t es un árbol binario no vacío que contiene números enteros.
// Post: Devuelve la suma máxima posible de un camino entre dos nodos
//       cualquiera del árbol dado. El camino de suma máxima puede
//       pasar o no por la raíz.
int maxSumPath(BinaryTree<int> t);
```

Aquí tenéis dos ejemplos de entrada de la función y sus correspondientes salidas. El primero, el camino de suma máxima pasa por la raíz y el segundo no pasa por la raíz:

```
2 (4 (7, 6), 5 (3, 4))
=>
22 -> 7+4+2+5+4
1 (1 (2, 3), 5 (9, 9))
=>
23 -> 5+9+9
```

Fijaos que el enunciado de este ejercicio ya os ofrece unos ficheros que lo debéis de utilizar para compilar: `Makefile`, `program.cpp`, `BinaryTree.hpp`, `maxSumPath.hpp`. Os faltará crear el fichero `maxSumPath.cpp` con los correspondientes `includes` y `implementar` la función anterior. Cuando vayáis a entregar vuestra solución en el JUTGE, sólo es necesario que entreguéis un **TAR** construido así:

```
tar cf solution.tar maxSumPath.cpp
```

Entrada

La entrada tiene un número arbitrario de casos. Cada caso consiste en una línea con un string describiendo un árbol binario de enteros. Fijaos en que el programa que ya os dan se encarga de leer estas entradas. Solo es necesario que implementéis la función anteriormente citada.

Salida

Para cada caso, es necesario escribir la suma máxima de un camino entre dos nodos cualquiera del árbol binario. Fijaos en que el programa que se os da ya se encarga de escribir estas salidas. Solo es necesario que implementéis la función anteriormente citada.

Ejemplo de entrada 1

```
1 (2 (-4), 3 (5 (7, 8), 6))
1 (2, 3)
5 (-1 (11 (-1, 2), ), 8 (13, 7 (-1)))
```

```
3 (4 (-10, 4), 5)
-15 (5 (-8 (2, -3), 1), 6 (3, 9 (0 (4, -1 (10, )))))
-10 (9, 20 (15, 7))
1 (3, 7 (9, ))
```

2 (1 (3, 7 (9,)), 3)

Ejemplo de salida 1

22
6
38
16
27
42
20
22

Ejemplo de entrada 2

3 (1 (4 (0, 2), 2 (1, 0)), 2 (1 (4, 5), 3 (5, -1)))
0 (1 (1, 2), -1 (4, 1))
4 (4 (5 (0 (4, 4), -1 (2, 4)), 2 (1 (0, -1), 1 (1, 0))), 3 (4 (-1 (5, 2), 5 (1, 1)), 0 (2 (4, -1), 0 (0, 1))))
3 (0 (2 (1 (5, 3), 4 (2, 4)), 4 (3 (2, -1), 4 (2, 1))), 2 (2 (1 (3, -1), 1 (-1, -1)), 2 (4 (-1, 3), 3 (-1, 1))))
0 (4 (3 (4 (-1, -1), 0 (3, 5)), 3 (-1 (3, -1), 2 (-1, 0))), 2 (4 (-1 (5 (2, -1), 5 (0, 3)), 1 (5 (5, -1), 5 (5, 0))))
2 (2 (3 (2, 3), 2 (4, 0)), 4 (3 (2, 4), 4 (0, -1)))
0 (3 (4 (-1, 2), 3 (3, 2)), 3 (2 (2, 2), 2 (2, 1)))
3 (3 (4, 5), 4 (3, -1))
0 (1 (-1 (5 (2, -1), 5 (4, 1)), 3 (2 (2, 0), -1 (-1, 2))), 1 (4 (0, -1), 1 (-1, 3)))
1 (2 (4, 4), 2 (3, 0))

Ejemplo de salida 2

20
6
33
24
27
21
16
18
14
12

Observación

Vuestra función y subfunciones que creáis han de trabajar solamente con árboles. Debeis de encontrar una solución **RECURSIVA** del problema. En la llamadas recursivas, incluye la hipótesis de inducción, es decir, una explicación de lo que se cumple después de la llamada y también la función de fita/decrecimiento o una justificación de porqué la función recursiva acaba.

Una solución directa superará los juegos de pruebas públicos y le permitirá obtener una nota razonable. Pero muy posiblemente será lenta, y necesitará crear alguna función recursiva auxiliar para producir una solución más eficiente capaz de superar todos los juegos de pruebas.

Evaluación sobre 10 puntos:

- Solución lenta: 7 puntos.
- Solución lenta + justificación: 8 puntos.
- solución rápida: 9 puntos.
- solución rápida + justificación: 10 puntos.

Entendemos como solución lenta una que es correcta y capaz de superar los juegos de pruebas públicos. Entendemos como solución rápida una que es correcta y capaz de superar los juegos de pruebas públicos y privados. La justificación vale 1 punto y consiste en definir correctamente las PRE/POST de las funciones auxiliares que añada y en definir correctamente las hipótesis de inducción y funciones de fita.

Información del problema

Autor : STUDENTS PRO1

Generación : 2022-12-24 02:11:04

© *Jutge.org*, 2006–2022.
<https://jutge.org>