

TAREA I: Análisis de Algoritmos

Prof. Andrea Rodríguez, Ayudante: Luis Cabrera

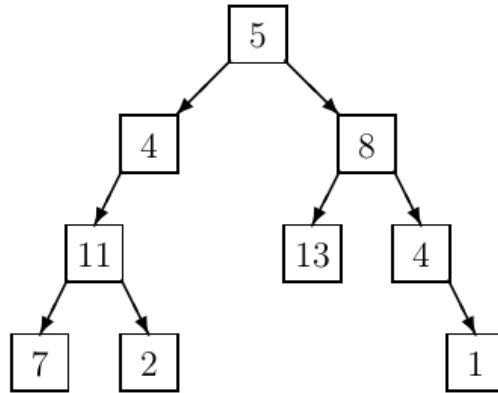
GRUPO 6

Los grupos deberán resolver un problema que ha sido asignados en forma aleatoria. Este problema consiste en diseñar e implementar un algoritmo. Se pide demostrar que el algoritmo es correcto, analizar el costo asintótico del algoritmo, y hacer una evaluación experimental en base a entradas variables de tamaño. Se debe entregar un informe y el código fuente que sea compilable en linux (C o C++) . El trabajo se debe entregar el día lunes 17 de abril a las 18:00 en forma digital por correo a la profesora y las presentaciones de los trabajos se harán a partir de la semana del 20 de abril.

Los árboles son fundamentales en muchas ramas de la informática. Este problema implica construir y recorrer árboles binarios.

Dada una secuencia de árboles binarios, debes escribir un programa que imprime un recorrido de orden de nivel de cada árbol. En esto problema cada nodo de un árbol binario contiene un entero positivo y todos los árboles binarios tienen menos de 256 nodos. En un recorrido de orden de nivel de un árbol, los datos en todos los nodos en nivel dado se imprimen en orden de izquierda a derecha y todos los nodos en nivel k se imprimen antes de todos los nodos en el nivel $k + 1$.

Por ejemplo, un recorrido de orden de nivel del árbol a la derecha en la figura es: 5, 4, 8, 11, 13, 4, 7, 2, 1.



En este problema un árbol binario es especificado por una secuencia de pares '(n, s)', donde n es el valor en el nodo cuya trayectoria desde la raíz viene dada por la cadena s . Un camino es dado ser una secuencia de 'L' y 'R' donde 'L' indica una rama izquierda y 'R' indica una rama derecha. En el árbol representado anteriormente, el nodo que contiene 13 es especificado por (13, RL), y el nodo que contiene 2 es Especificado por (2, LLR). El nodo raíz se especifica mediante (5,) donde la cadena vacía indica la ruta desde la raíz a sí mismo. Se considera que un árbol binario se especifica completamente o es completo si cada nodo tiene un valor y solo uno. Como ejemplo, considere la siguiente entrada con 2 árboles

```
(11,LL) (7,LLL) (8,R) (5,) (4,L) (13,RL) (2,LLR) (1,RRR) (4,RR) ()
(3,L) (4,R) ()
```

Para cada árbol binario especificado en el archivo de entrada, el recorrido de orden de nivel de ese árbol debe ser impreso. Si un árbol no está completamente especificado, escribir no completo.

Por ejemplo, considere la siguiente entrada:

```
(11,LL) (7,LLL) (8,R) (5,) (4,L) (13,RL) (2,LLR) (1,RRR) (4,RR) ()
(3,L) (4,R) ()
```

La salida es entonces

```
5 4 8 11 13 4 7 2 1
```

no completo