

1. **El árbol que se crea a partir de los ficheros de nodos (*dict*.dat*), ¿es completo o casi completo? Justifica tu respuesta.**

Es casi completo, ya que no se busca una forma de insercción ordenada en la implementación. Se podría dar que alguna de las posibles entradas generase un árbol completo pero no sería de forma intencionada.

2.

- a) **¿Qué relación hay entre la “forma” de un árbol y sus recorridos?**

Dependiendo de la forma que tiene será mas eficiente un tipo de recorrido u otro. Si generásemos un árbol binario de búsqueda y lo balanceáramos en la insercción, podríamos usar diferentes algoritmos para encontrar uno de los nodos dependiendo de su posición. No obstante, la insercción es secuencial y, aunque el árbol guarda cierto orden por ser un ABdB, no podemos asegurar que los algoritmos conocidos puedan ser óptimos.

- b) **Se puede saber si un árbol binario de búsqueda está bien construido según sus recorridos?**

Si, dado que, en caso de estar mal contruido, ciertos recorridos reproducirían incoherencias. Por ejemplo, el recorrido medio debería devolver una lista ordenada si el árbol está bien construido.

3. **Compara y describe las diferencias entre los árboles generados por los ejecutables *p4_e3* con el último argumento *B* o *N* (número de nodos, profundidad, recorridos, etc.).**

En la siguiente captura podemos ver la diferencia entre una ejecución y otra con el diccionario de 1M.

La principal diferencia es el tiempo medido en la ejecución: la ejecución balanceada tarda la mitad que la otra.

Por otra parte, la profundidad del árbol también es diferente, lo cual está directamente relacionado con el punto anterior

El recorrido de la ejecución balanceada muestra un orden correcto mientras que la otra ejecución muestra un orden, aparentemente, aleatorio

```
roman@SM-ROMAN:/mnt/c/Users/roman/Documents/workspace/Prog2$ ./p4_e3 inputs/dict1M.dat B
1000000 lineas leidas
Datos ordenados

Tiempo de creacion del Arbol: 1062500 ticks (1.062500 segundos)
Numero de nodos: 1000000
Profundidad: 20
Introduce un nodo para buscar en el Arbol (siguiendo el mismo formato que en el fichero de ent
^[[A^C
roman@SM-ROMAN:/mnt/c/Users/roman/Documents/workspace/Prog2$ ./p4_e3 inputs/dict1M.dat N
1000000 lineas leidas

Tiempo de creacion del Arbol: 2265625 ticks (2.265625 segundos)
Numero de nodos: 1000000
Profundidad: 49
Introduce un nodo para buscar en el Arbol (siguiendo el mismo formato que en el fichero de ent
█
```