
ESTRUCTURAS DE DATOS

TP 3: Recursión - Árboles

1. Escriba una función que calcule el factorial de un número en el lenguaje que prefiera. Implemente una solución iterativa y una recursiva. Compare ambas soluciones e indique ventajas y desventajas de cada una.
2. Visite el sitio www.cs.usfca.edu/~galles/visualization/RecReverse.html y analice la solución propuesta para obtener la palabra reversa. Visualice usando esta herramienta la ejecución de la función reverse. Proponga otras alternativas de solución. ¿Puede usar el TAD Pila implementado en el Trabajo Práctico 2?
3. Compare todas las alternativas propuestas en el ejercicio 2 indicando ventajas y desventajas de cada una.
4. Implemente una solución recursiva para el método copy() de la pila implementada en el ejercicio 1 del Trabajo Práctico 2. ¿Encuentra alguna ventaja en esta nueva alternativa de la implementación?
5. Implemente de manera recursiva el método __len__ del TAD Pila (implementada con nodos) visto en la teoría. ¿Encuentra alguna ventaja en esta nueva alternativa de la implementación?
6. Implemente una función que sume los números que contiene una lista de Python. Implemente una solución iterativa y una recursiva. Compare ambas soluciones e indique ventajas y desventajas de cada una.
7. Analice las siguientes funciones:

```
def fib1(n):
    if n==0:
        return 0
    if n==1:
        return 1
    return fib1(n-1) + fib1(n-2)
```

```
def fib2(n):
    def do_fib((a, b, i)
        if i == n:
            return a
        return do_fib(b, a+b, i+1)

    return do_fib(0, 1, 0)
```

¿Qué opina sobre las soluciones propuestas?
Compárelas y luego indique cual recomendaría utilizar.

8. TAD Árbol Binario de Búsqueda.
 - a) Especifique el TAD.
 - b) Indique con sus palabras qué casos debe considerar en el algoritmo de inserción y qué casos en el algoritmo de borrado de una clave en el árbol.
 - c) Implemente el TAD usando Python detalle las decisiones tomadas para la elección de la estructura interna.
 - d) Para cada operación implementada indique que opción resulta más conveniente, justificando: solución “recursiva” o “iterativa”.

- e) Un recorrido posible para un árbol es el recorrido en orden. ¿Qué otros recorridos posibles puede proponer?
 - f) Implemente una Coordenada de avance para cada una de las propuestas que hizo en d)
 - g) Si no lo hizo aún agregue lo necesario a la implementación para que el ABB sea iterable.
9. TAD Árbol Balanceado AVL.
- a) Especifique el TAD.
 - b) Indique con sus palabras que casos debe considerar para mantener el AVL balanceado luego de agregar un nuevo nodo en el árbol. Indique que rotaciones debería hacer para revalancear el árbol en cada uno de los casos. En el proceso de rebalanceo del árbol, ¿ayuda que el algoritmo sea recursivo? Justifique su respuesta.
 - c) Repita lo pedido en b) para el caso de borrar un nodo.
 - d) Implemente el TAD usando Python. Detalle las decisiones tomadas para la elección de la estructura interna. Para cada operación indique que opción resulta más conveniente, justificando: solución “recursiva” o “iterativa”.
 - e) ¿Qué recorridos puede sugerir usando una Coordenada para un árbol AVL? Implemente los recorridos propuestos.
 - f) ¿Cómo implementaría este TAD si se decidiera no almacenar la altura del subárbol en cada nodo? ¿En qué situación tendría sentido hacerlo? Piense en un entorno restringido en memoria RAM
 - g) Si no lo hizo agregue lo necesario a la implementación para que el AVL sea iterable.
10. Implemente el TAD conjunto incorporando las mejoras que pueda hacer ahora usando árboles. Justifique las mejoras en la implementación. ¿Debe modificar algo en la especificación hecha en el Trabajo Práctico 1?