2018

Erick Efrain Vargas Romero

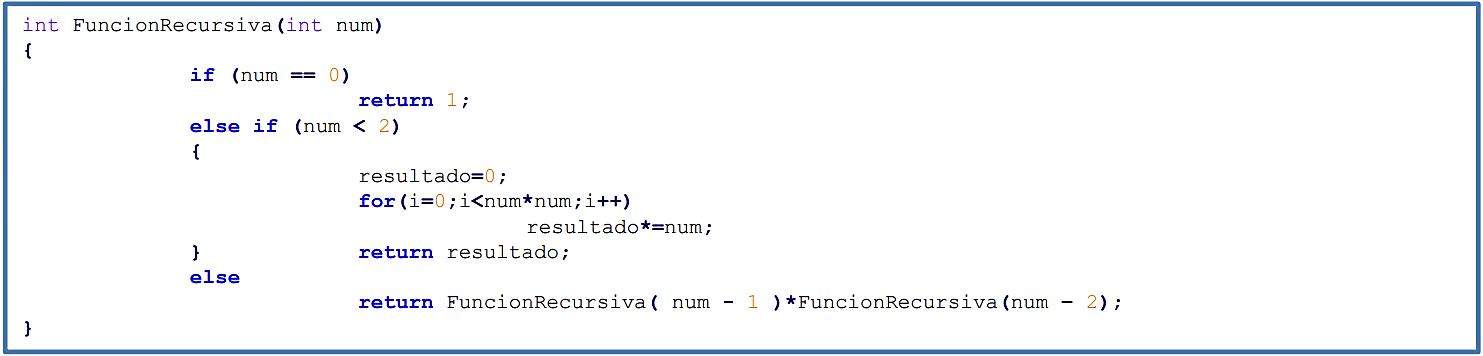
Prof. Franco Martínez Edgardo Adrián

Análisis de algoritmos recursivos

3CM2

Análisis de algoritmos



1. Calcular la cota de complejidad para el algoritmo de la siguiente función recursiva

T(0) = 1

T(n-1) + T(n-2) + 1

Si consideramos los return y llamadas a función como operaciones básicas obtenemos

Por tanto

Si calculamos C1,C2 y C3 utilizando T(0), T(1) y T(2) obtenemos que

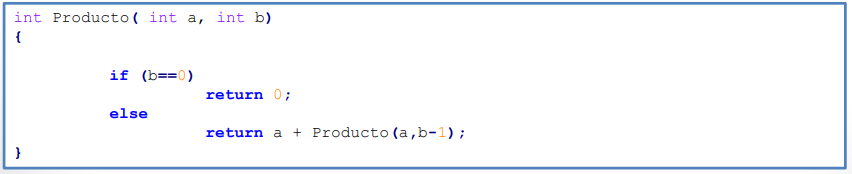
T(0) = C1=1

T(1) =

T(2)=

Finalmente

1. Calcular la complejidad de la implementación recursiva del producto



T(n-1) + 1

1

Si consideramos los return y llamadas a función como operaciones básicas obtenemos

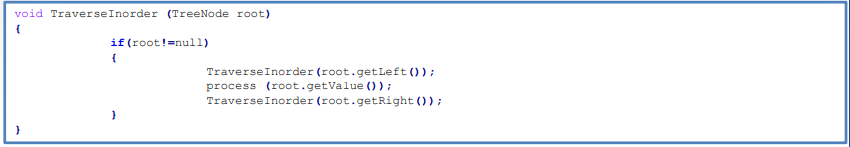
Por tanto

Utilizando la condición inicial T(0) = 1

Calculando los coeficientes obtenemos que

Finalmente

1. Calcular el costo de un recorrido en in-orden de un árbol binario completamente lleno



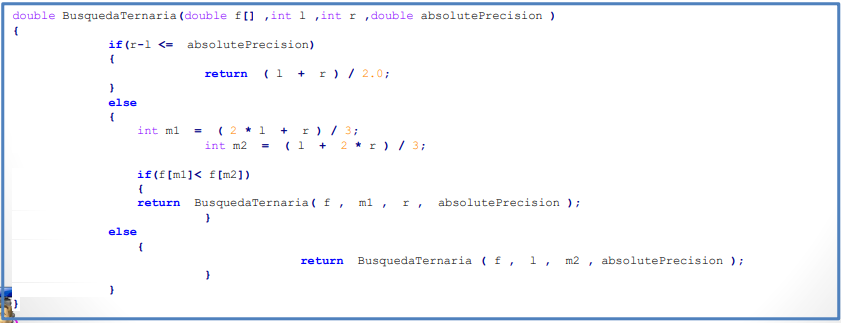
1

Si consideramos las llamadas a función y la comparación como operaciones básicas obtenemos

Debemos aplicar teorema maestro para llegar a la solución correcta

Por tanto, usando caso dos del teorema maestro

1. Calcular la cota de complejidad de un algoritmo de búsqueda ternaria

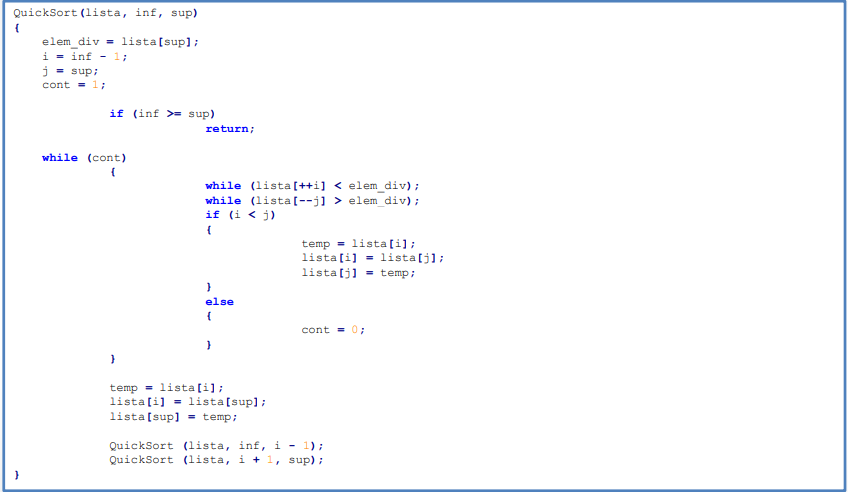


1

Si consideramos las llamadas a función y la comparación como operaciones básicas obtenemos

Por tanto debemos aplicar teorema maestro para poder obtener la cota del algoritmo de búsqueda ternaria

1. Calcular la cota de complejidad del algoritmo de ordenamiento quick sort



n

1

1

Nuevamente nos apoyaremos del teorema maestro para la resolución de este problema, considerando como operaciones básicas la primer condición que se realiza y los returns

1. Resolver las siguientes ecuaciones y dar su orden de complejidad:
2. Calcular la cota de complejidad que tendrían los algoritmos con los siguientes modelos recurrentes