

Algoritmos

TAREA 4

Israel Sandoval Grajeda y Fausto Salazar Mora

5 de diciembre de 2013

Divide y vencerás

Algoritmo para la multiplicación de bits.

Para la ejecución del algoritmo presentado a continuación se deben realizar las siguientes consideraciones:

1. a y b son arreglos de bits
2. Su longitud es 2^n

Algorithm 1 Multiplicación de bits

```
1: MultBits(a,b)
2: n=longitud(a)/2
3: if n==1 then
4:   return a*b
5: end if
6: a2=a[n:]
7: b1=b[:n]
8: b2=b[n:]
9: z2=MultBits(a2,b2)
10: z0=MultBits(a1,b1)
11: z1=MultBits(a1+a2,b1+b2)-z2-z0
12: ShiftL(z2,n)
13: ShiftL(z1,n/2)
14: return z2+z1+z0
```

Complejidad:

En la función ShiftL hay un ciclo hasta n para hacer los corrimientos, las demás operaciones en esta función son constantes así pues el tiempo es lineal sobre el tamaño de n es decir $O(n)$

En la recursión todas las operaciones son constantes y la recursión tiene la forma de $3 T(n/2)$ entonces la complejidad de la recursión es $T(n) = 3 T(n/2) + O(n)$. Aplicando el método maestro tengo $a=3$ $b=2$ y $f(n) = O(n)$ comparando $f(n)$ con $n^{\log_b a}$ cae en el caso 1 del método maestro y por lo tanto $T(n) = O(n^{\log_2 3})$

Correctez:

El predicado siguiente indica que es correcto:

En cada llamada recursiva de la función, el tamaño de las entradas disminuye a la mitad y al volver de la recursión siempre me regresa el resultado de multiplicar las entradas.