

Marcos Geraldo Braga Emiliano

19.1.4012

1. Usando o algoritmo que multiplica dois números binários a um custo $\Theta(n^{1.585})$, multiplique 1001 por 0110.

2. Suponha que você deseja escolher um dentre os três algoritmos:

a) Algoritmo A soluciona problemas por dividi-los em cinco subproblemas com metade do tamanho, recursivamente soluciona cada subproblema e combina as soluções em tempo linear.

b) Algoritmo B soluciona problemas de tamanho n por, recursivamente, solucionar 2 subproblemas de tamanho $n-1$ e combinar as soluções em tempo constante.

c) Algoritmo C soluciona problemas de tamanho n dividindo-os em 9 subproblemas de tamanho $n/3$, recursivamente soluciona cada subproblema e, então, combina as soluções em tempo quadrático.

Qual é o tempo de execução de cada algoritmo em notação O e qual você deveria escolher?

1. Multiplicação de 1001 por 0110:

Passo 1) 1001×0110

Passo 2) $10 \times 01 \mid 10 \times 01 \mid (10+01) == 11 \times (01+10) == 11$

Passo 3.1(10×01) $\Rightarrow 1 \times 0 \mid 0 \times 1 \mid (1+0) \times (0+1)$

Passo 3.2(10×01) $\Rightarrow \mid 0 \times 1 \mid 1 \times 0 \mid (0+1) \times (1+0)$

Passo 3.3(11×11) $\Rightarrow 1 \times 1 \mid 1 \times 1 \mid (1+1) \times (1+1)$

Casos base:

Retorno 4.1) $0 \mid 0 \mid 1$

Retorno 4.2) $0 \mid 0 \mid 1$

Retorno 4.3) $1 \mid 1 \mid 1$

Retorno 5.1) $0 * 2^2 + (1-0-0) * 2^{2/2} + 0 = 2$, pela regra $P1 * 2^n (P3-P1-P2) * 2^{n/2} + P2$

Retorno 5.2) $0 * 2^2 + (1-0-0) * 2^{2/2} + 0 = 2$

Retorno 5.3) $1 * 2^2 + (1-1-1) * 2^{2/2} + 1 = 3$

Retorno 6) $1 * 2^4 + (3-2-2) * 2^{4/2} + 2 = 14$, pela regra $P1 * 2^n (P3-P1-P2) * 2^{n/2} + P2$

2. a) $T(n) = 5 * T(n/2) + O(n)$, temos que $n = O(n^{\log_2 5})$, pois pelo teorema mestre $\log_2 5 > 1$, desta forma $n = O(n^{\log_2 5})$

b) $T(n) = 2 * T(n-1) + O(1)$, temos que:

$T(n) = 2 * T(n-1) + O(1) \Rightarrow$

$$T(n-1) = 2*(2T(n-2) + O(1)) + O(1) \Rightarrow$$

$$T(n-2) = 2*(2(2T(n-3) + O(1)) + O(1)) + O(1) \Rightarrow$$

$$T(n-3) = 2*(2(2(2T(n-4) + O(1)) + O(1)) + O(1)) + O(1) \Rightarrow$$

.

.

$$T(n) = 2^k * T(n-k) + 2^k O(1) \Rightarrow$$

#Não consegui terminar e encontrar a relação de recorrência

c) $T(n) = 9 * T(n/3) + O(n^2)$, temos que $n = O(n^2 \log n)$, pois pelo teorema mestre $\log_3 9 == 2$, desta forma $n = O(n^2 \log n)$

Dentre os que encontrei temos como melhor desempenho $n = O(n \log 25)$