Lista de exercícios 2: Divisão-e-conquista
Entrega: 26/03/2012

Exercício 1
Resolve as seguintes recorrências com o método de Akra-Bazzi ou justifica porque o método não se aplica. Caso o método de Akra-Bazzi não se aplica, resolve usando outro método. Dê uma cota Θ para cada uma delas.

a) $T(n) = 2T(n/3) + 1$

b) $T(n) = 5T(n/4) + n$

c) $T(n) = 7T(n/7) + n$

d) $T(n) = 9T(n/3) + n^2$

e) $T(n) = 8T(n/2) + n^3$

f) $T(n) = 49T(n/25) + n^{3/2} \log n$

g) $T(n) = T(n-1) + 2$

h) $T(n) = T(n-1) + n^c$, com constante $c \geq 1$

i) $T(n) = T(n-1) + c^n$, com constante $c > 1$

j) $T(n) = 2T(n-1) + 1$

k) $T(n) = T(\sqrt{n}) + 1$

Exercício 2
Pratique com a multiplicação de polinômios usando a transformação rápido de Fourier (TRF).

a) Suponha que você queira multiplicar os dois polinômios $x + 1$ e $x^2 + 1$ usando TRF. Escolha uma potência de dois apropriada, encontra a TRF das duas sequências, multiplique os resultados por componente, e compute a TRF inversa para obter o resultado final.

b) Repita para o par de polinômios $1 + x + 2x^2$ e $2 + 3x$.

Exercício 3
O *quadrado* de uma matriz A é seu produto consigo mesma, AA.

a) Mostre que cinco multiplicações são suficientes para computar o quadrado de uma matriz 2×2.

b) O que está errado com o seguinte algoritmo para computar o quadrado de uma matriz 2×2?

> "Use uma abordagem de divisão-e-conquista como no algoritmo de Strassen, exceto que em vez de obter 7 subproblemas de tamanho $n/2$, agora obtém 5 subproblemas de tamanho $n/2$ graças ao item (a). Usando a mesma análise do algoritmo de Strassen, podemos concluir que o algoritmo executa em tempo $O(n^{\log_2 5})$."

c) De fato, elevar matrizes ao quadrado não é mais fácil do que multiplicar matrizes. Mostra que se matrizes $n \times n$ podem ser elevadas ao quadrado em tempo $O(n^c)$, então quaisquer duas matrizes $n \times n$ podem ser multiplicadas em tempo $O(n^c)$.

Exercício 4
As *matrizes de Hadamard* $H_0, H_1, \ldots$ são definidas por

- H_0 é a matriz $(1) \in \mathbb{R}^{1\times 1}$

- Para $k > 0$, a matriz $H_k \in \mathbb{R}^{2^k \times 2^k}$ é

$$H_k = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{pmatrix}$$

Mostre que se v é um vetor coluna de tamanho $n = 2^k$, então o produto matriz-vetor $H_k v$ pode ser calculado usando $O(n \log n)$ operações. Considere que todos os números envolvidos sejam pequenos o suficiente para que operações básicas como adição e multiplicação tomem tempo unitário.

Exercício 5
Supõe que temos um algoritmo para multiplicar dois inteiros binários de n bits x e y em tempo n^a, onde $a = \log_2 3$ (ver as notas de aula para um algoritmo desse tipo). Chame esse procedimento de multiplicação_rápido(x, y).

a) Queremos converter o inteiro decimal 10^n (um 1 seguindo de n zeros) para binário. Aqui está o algoritmo (considere que n é uma potência de 2):

```
função pot2bin(n)
  se n=1: retorna 1010₂
  senão:
    z = ???
    retorna multiplicação_binário(z,z)
```

Complete os detalhes que estão faltando. Depois forneça uma relação de recorrência para o tempo de execução do algoritmo e resolva a recorrência.

b) Depois, queremos converter qualquer inteiro decimal x com n dígitos (onde n é uma potência de 2) para binário. O algoritmo é o seguinte:

```
função dec2bin(x)
  se n=1: retorna binário[x]
  senão:
    divida x em dois números decimais x_L,x_R com n/2 dígitos cada
    retorna ???
```

Aqui, binário$[x]$ é um vetor que contém a representação binário de todos os inteiros de um dígito. Ou seja, binário$[0]=0_2$, binário$[1]=1_2$, até binário$[9]=1001_2$. Considere que um acesso binário toma tempo $O(1)$. Complete os detalhes ausentes. Mais uma vez, forneça uma recorrência para o tempo de execução do algoritmo e a resolva.