Construção e Análise de Algoritmos

lista de exercícios 12

1. **Multiplicação de inteiros mais eficiente ainda (?)**

   Imagine que alguém decide reduzir ainda mais o tempo da multiplicação de inteiros dividindo os números em 3 partes ao invés de 2.

   $$X \quad = \quad X_3 \cdot 10^{2n/3} \ + \ X_2 \cdot 10^{n/3} \ + \ X_1$$

   Qual o número máximo de multiplicações de números com $n/3$ dígitos que ela pode realizar, para que o seu algoritmo seja mais eficiente do que aquele apresentado na aula?

   Você consegue fazer as coisas funcionarem dessa maneira?

2. **O método de Euclides**

   O método de Euclides para o cálculo do MDC de dois números a e b é baseado na seguinte observação

   $$MDC(a,b) \quad = \quad MDC(b,r)$$

   onde r é o resto da divisão de a por b, e nós estamos assumindo que $a > b$.

   Note que essa estratégia para o cálculo do MDC nos dá mais um exemplo de aplicação da técnica de divisão e conquista.

   Escreva o pseudo-código do algoritmo que calcula o MDC de a e b pelo método de Euclides.

   Estime o tempo de execução do seu algoritmo, assumindo que a e b são números com n dígitos.

3. **Outro método para o cálculo do MDC**

   Nesse exercício, nós vamos ver uma estratégia alternativa de divisão e conquista para o cálculo do MDC.

   a) Verifique que a seguinte recorrência para o MDC é válida

   $$\mathsf{mdc}(a,b) \quad = \quad \begin{cases} 2 \cdot \mathsf{mdc}(a/2, b/2) & \text{, se } a \text{ e } b \text{ são ambos pares} \\ \mathsf{mdc}(a, b/2) & \text{, se } a \text{ é ímpar e } b \text{ é par} \\ \mathsf{mdc}((a-b)/2, b) & \text{, se } a \text{ e } b \text{ são ambos ímpares} \end{cases}$$

   b) Apresente um algoritmo de divisão e conquista para o cálculo do MDC baseado nessa recorrência.

c) Estime o tempo de execução desse algoritmo, assumindo que a e b são números com n dígitos.

4. **Convolução**

A distribuição de probabilidades de uma variável aleatória discreta X pode ser representada na forma de um vetor como

$$f_X \quad = \quad \big(p_0, p_1, p_2, \ldots, p_n\big)$$

onde $p_k = P(X = k)$.

Por exemplo, se X é o número de Caras obtidas no lançamento de 4 moedas comuns, então a sua distribuição de probabilidades é

$$f_X \quad = \quad \Big(\frac{1}{16}, \frac{4}{16}, \frac{6}{16}, \frac{4}{16}, \frac{1}{16}\Big)$$

Agora, suponha que X e Y são variáveis aleatórias independentes, com distribuições de probabilidades

$$f_X \quad = \quad \big(p_0, p_1, p_2, \ldots, p_n\big) \qquad\qquad f_Y \quad = \quad \big(q_0, q_1, q_2, \ldots, q_n\big)$$

Então, $X + Y$ também é uma variável aleatória, com distribuição de probabilidades

$$f_{X+Y} \quad = \quad \big(r_0, r_1, r_2, \ldots, r_{2n}\big)$$

onde cada probabilidade r_k é calculada da seguinte maneira

$$r_k \quad = \quad \sum_{i+j=k} p_i \cdot q_j$$

A operação que produz o vetor f_{X+Y} a partir dos vetores f_X e f_Y é conhecida como *convolução*.

A implementação ingênua dessa operação, de acordo com a equação acima, executa em tempo $\Theta(n^2)$.

Apresente um algoritmo de divisão e conquista que calcula a convolução de dois vetores de maneira mais eficiente do que isso.