

Tópicos Avançados em Algoritmos - Programação Dinâmica I

1. Optimizar o produto de matrizes

Considere o problema da parentização óptima de uma multiplicação de matrizes.

- (a) Escreva a equação fundamental de minimização em que se baseia a aplicação da Programação Dinâmica a este problema. Defina as variáveis que usar.
- (b) Usando a versão “bottom-up” da Programação Dinâmica, quantos sub-problemas¹ são resolvidos? Nota. Só se consideram os sub-problemas que envolvem 2 ou mais matrizes. Mostre que se obtém uma eficiência de ordem $O(n^3)$.
- (c) Resolva o seguinte problema usando a versão “bottom-up” da Programação Dinâmica. Quantos sub-problemas são resolvidos?

produto $M_2M_2M_3M_4$, $M_1[3 \times 10]$, $M_2[10 \times 100]$, $M_3[100 \times 20]$, $M_4[20 \times 5]$

2. Números de Catalan e colocação de parêntesis

O número de modos de colocar parêntesis num produto de n factores é o número² de Catalan c_{n-1} . Por exemplo, para $n = 4$, temos 5 modos

$$a(b(cd)), a((bc)d), (ab)(cd), (a(bc)d), ((ab)c)d$$

de modo que $c_3 = 5$. De forma análoga se vê que $c_2 = 2$ e $c_1 = 1$. Indique uma recorrência que exprime c_n para $n \geq 2$ como função de valores c_i para $i < n$.

Usando essa recorrência, escreva numa linguagem de programação apropriada uma função que calcula c_n .

3. Optimizar o produto de inteiros

Considere o problema de determinar a forma óptima de multiplicar n inteiros $a_1a_2 \dots a_n$. Modelo de custos. O custo da multiplicação ab é $|a| \cdot |b| \approx \log a \log b$. Estude o efeito da parentização ($(ab)c$ ou $a(bc)$) no valor do custo do produto. Estude o efeito da utilização da propriedade comutativa no valor do custo do produto.

¹Isto é, cálculos de parentizações óptimas.

²Ver por exemplo “O Livro dos Números” de John Conway e Richard Guy, Gradiva, 1999.

4. Edição de texto e máxima sequência (não adjacente) comum

Considere a transformação de uma string s numa string t . Seja $m = |s|$ e $n = |t|$. Mostre que o número mínimo de operações de edição do tipo “inserir character” ou “eliminar character” é dado por

$$m + n - 2 \times \text{maxsub}(s, t)$$

onde $\text{maxsub}(s, t)$ é o comprimento da maior sub-sequência comum às strings s e t .

5. O problema da mochila é completo na classe NP³

Mostre que o problema da mochila, na sua versão de decisão, é completo em NP.

INSTÂNCIA: (Problema da mochila) Conjunto $A = \{1, 2, \dots, n\}$, tamanho t_i para cada objecto de A , valor v_i para cada objecto de A , inteiros positivos T e V .

PERGUNTA: Existe $B \subseteq A$ tal que

$$\left[\sum_{i \in B} v_i \geq V \right] \wedge \left[\sum_{i \in B} t_i \leq T \right]$$

Sugestão. Efectue uma redução polinomial $\text{PART} \leq_p \text{Mochila}$.

INSTÂNCIA: (Problema da PART) Conjunto $A = \{1, 2, \dots, n\}$, tamanho s_i para cada objecto de A .

PERGUNTA: Existe uma partição $A = A_1 \cup A_2$ tal que $\sum_{i \in A_1} s_i = \sum_{i \in A_2} s_i$?

Nota. Tem mais exemplos e exercícios de Programação Dinâmica no capítulo “Dynamic Programming” do livro de Cormen, Charles E. Leiserson e Ronald L. Rivest, *Introduction to Algorithms* (second edition) MIT Press and McGraw-Hill, 2001.

³Para resolver este problema é necessário ter alguns conhecimentos da teoria dos problemas NP-completos – tema que normalmente faz parte do programa da disciplina de Complexidade.