Programação Dinâmica

programação remete a planejamento

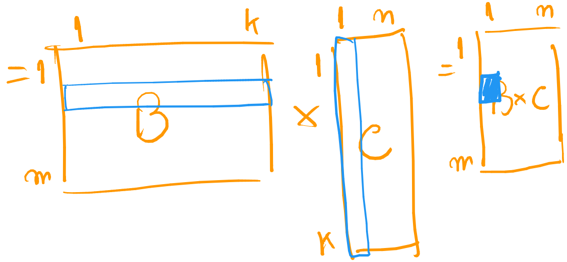
- Entender problema
- Imaginar sol. ótima
- Mostrar subestrutura ótima
- Deduzir recorrência
- Projetar alg. c/ tabela
- Analisar efic. de tempo e espaço

Multiplicação de Cadeias de Matrizes

Entrada: uma seq. de matrizes $A_1, A_2, \dots, A_n$ para multiplicar, c/ A_i tendo tamanho $m_i \times m_{i+1}$ p/ $i = 1$ até n (note que m_{n+1} é o # de cols. de A_n)

Solução: a seq. de menor custo p/ obter $A_1 \times A_2 \times \dots \times A_n$

Note que $B_{m \times k} \times C_{k \times n}$ =



custa $\underline{m.k.n}$

Exemplo

Como multiplicação de matrizes é associativa, temos diferentes formas para

$A_{50\times 20}$

$B_{20\times 1}$

$C_{1\times 10}$

$D_{10\times 100}$

$$A \times B \times C \times D$$

i) $((A\times(B\times C))\times D)$ custa $20.1.10 + 50.20.10 + 50.10.100 =$

$= 200 + 10000 + 50000 = 60200$

ii) $((A\times B)\times(C\times D))$ custa $50.20.1 + 1.10.100 + 50.1.100 =$

$= 1000 + 1000 + 5000 = 7000$

iii) $(A\times(B\times(C\times D)))$ custa $= 1.10.100 + 20.1.100 + 50.20.100 =$

$= 1000 + 2000 + 100000 = 103000$

Quiz: de quantas maneiras podemos multiplicar n matrizes (em ordem)? Resp.: $(n-1)!$, mas tem redundâncias

Imaginar Solução Ótima

⊛ alg. guloso pode parecer promissor, mas não funciona sempre

- Note que nossa escolha é apenas a ordem das multiplicações, já que toda sol. faz $n-1$ produtos

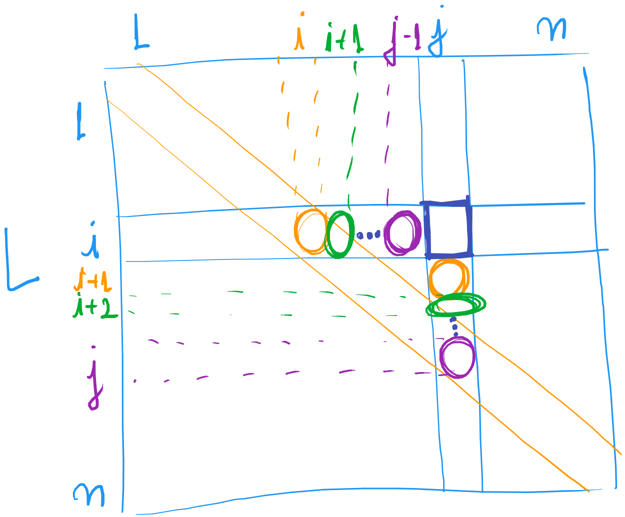
- Seja $L[1,n]$ a solução ótima p/ multiplicar as matrizes $A_1 \times \ldots \times A_n$ e suponha que o último produto foi entre as posições K e $K+1$

$$L[1,n] = m_1 \cdot m_{K+1} \cdot m_{n+1} + L[1,K] + L[K+1,n]$$

⊛ Note que temos subestrutura ótima, i.e., $L[1,K]$ e $L[K+1,n]$ são sols. ótimas de seus subproblemas, caso contrário $L[1,n]$ não é ótima

Deduzir Recorrência

$$L[i,j] = \min_{k=i..j-1} \left\{ m_i \cdot m_{k+1} \cdot m_{j+1} + L[i,k] + L[k+1,j] \right\}$$



Caso Base:

$L[i,i] = \emptyset, \forall i = 1, \ldots, n$

Alg. iter. c/ tabela

Quiz: o que s representa? Resp.: o tam. do intervalo de matrizes.

multCadeiasMatrizes$(m[1..n+1])$:

$\Theta(n)$ — para $i = 1$ até n: $L[i,i] = 0$

para $s = 1$ até $n-1$:

 para $i = 1$ até $n-s$:

  $j = i+s$

  $L[i,j] = \min\limits_{k=i..j-1} \{ m_i \cdot m_{k+1} \cdot m_{j+1} + L[i,k] + L[k+1,j] \}$

devolva $L[1,n]$

$\Theta(n)$ iter.

$\Theta(n)$ iter.

$\Theta(n)$

efic. de tempo $\Theta(n^3)$

↳ Desafio: para reconstruir a sol., percorrer a tabela a partir de $L[1,n]$, verificando qual K minimizou e descendo recursivamente nos subproblemas $L[1,K]$ e $L[K+1,n]$

Efic. espaço: $\Theta(n^2)$ é o tam. da tabela L