Construção e Análise de Algoritmos

aula 12: Multiplicação eficiente de inteiros e matrizes

1 Introdução

Todos aprendem na escola a multiplicar inteiros da seguinte maneira

$$\begin{array}{r} 371 \\ \times\ 58 \\ \hline 2968 \\ 1855 \\ \hline 21518 \end{array}$$

Isto é, o primeiro número (todos os seus dígitos) é multiplicado por cada dígito do segundo número, e a seguir os resultados (deslocados uma casa cada um) são somados.

Assumindo que os dois números possuem n dígitos, não é difícil ver que esse procedimento leva tempo $\Theta(n^2)$.

Todos também aprendem na escola a multiplicar matrizes da seguinte maneira

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} (1 \cdot 5 + 2 \cdot 7) & (1 \cdot 6 + 2 \cdot 8) \\ (3 \cdot 5 + 4 \cdot 7) & (3 \cdot 6 + 4 \cdot 8) \end{bmatrix}$$

Isto é, cada linha da primeira matriz é multiplicada por uma coluna da segunda matriz para produzir uma entrada da matriz resultado.

Assumindo que as duas matrizes possuem dimensão $n \times n$, não é difícil ver que

- cada multiplicação de linha por coluna leva tempo $\Theta(n)$
- e, como são, n^2 entradas na matriz resultado, o procedimento leva tempo $\Theta(n^3)$

À primeira vista, não é fácil imaginar que essas operações possam ser executadas mais rápido que $\Theta(n^2)$ e $\Theta(n^3)$, respectivamente. Mas, a seguir nós vamos ver como isso pode ser feito.

2 Estratégia de decomposição

Considere primeiramente o problema da multiplicação de inteiros.

A primeira observação é que, ao invés de trabalhar com os dígitos individualmente, nós podemos, por exemplo, trabalhar com pares de dígitos

$$\begin{array}{r} (23)(71) \\ \times\ (16)(58) \\ \hline \end{array}$$

A ideia aqui é começar multiplicando o primeiro número (formado pelos pares 23 e 71) pelo par 58 — isso é a mesma coisa que trabalhar na base 100 (a aritmética da centopéia ...).

Ao multiplicar 58 por 71, nós obtemos 18 e "vai 41":[1]

$$\begin{array}{rr} & (41) \\ & (23)(71) \\ \times & (16)(58) \\ \hline & (18) \end{array}$$

A seguir, multiplicando 58 por 23, e somando 41, nós obtemos

$$\begin{array}{rr} & (23)(71) \\ \times & (16)(58) \\ \hline (13)(75)(18) \end{array}$$

Finalmente, repetindo a mesma operação para o par 16 e somando os resultados, nós obtemos a resposta

$$\begin{array}{rr} & (23)(71) \\ \times & (16)(58) \\ \hline & (13)(75)(18) \\ (3)(79)(36) & \\ \hline (3)(93)(11)(18) & \end{array}$$

Legal!

Agora nós estamos prontos para pensar sobre números bem grandes.

Considere a multiplicação abaixo, envolvendo dois números X e Y com n dígitos cada um

$$\begin{array}{rr} & \text{xn} \ \ldots \ \text{x2} \ \text{x1} \\ \times & \text{yn} \ \ldots \ \text{y2} \ \text{y1} \\ \hline \end{array}$$

A ideia dessa vez é trabalhar com grupos de $n/2$ dígitos, dividindo cada número exatamente no meio

$$\begin{array}{rr} & (\text{xn} \ \ldots \ \text{xn/2+1})(\text{xn/2} \ \ldots \ \text{x1}) \\ \times & (\text{yn} \ \ldots \ \text{yn/2+1})(\text{yn/2} \ \ldots \ \text{y1}) \\ \hline \end{array}$$

Denotando as duas partes de X e Y por X_2, X_1 e $Y2, Y1$, respectivamente, a coisa fica assim

$$\begin{array}{rr} & \text{X2} \ \text{X1} \\ \times & \text{Y2} \ \text{Y1} \\ \hline \end{array}$$

E agora, basta fazer as contas.

[1] Essa é a aritmética da centopéia ...

Observando que

$$X \;=\; X_2 \cdot 10^{n/2} \;+\; X_1 \qquad\qquad Y \;=\; Y_2 \cdot 10^{n/2} \;+\; Y_1$$

as contas podem ser feitas da seguinte maneira

$$\begin{aligned} X \cdot Y &= \big(X_2 \cdot 10^{n/2} \;+\; X_1\big) \times \big(Y_2 \cdot 10^{n/2} \;+\; Y_1\big) \\ &= X_2Y_2 \cdot 10^{n} \;+\; X_2Y_1 \cdot 10^{n/2} \;+\; X_1Y_2 \cdot 10^{n/2} \;+\; X_1Y_1 \end{aligned}$$

3 Uma estratégia de conquista esperta

Infelizmente, a estratégia de decomposição que nós acabamos de ver ainda não produz um resultado interessante.

Vejamos.

Nós já sabemos que a multiplicação de X e Y dígito a dígito leva tempo n^2.

De maneira análoga, as multiplicações envolvendo X_i,Y_j (que possuem $n/2$ dígitos cada um) levam tempo $\frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4}$.

Mas, como são realizadas 4 multiplicações desse tipo, o tempo total é de

$$4 \cdot \frac{n^2}{4} \;=\; n^2$$

Ou seja, nós não ganhamos nada até aqui.

A observação chave, a seguir, é que a multiplicação de X e Y também pode ser escrita como

$$X \cdot Y \;=\; X_2Y_2 \cdot 10^{n} \;+\; \big(X_2Y_1 \;+\; X_1Y_2\big) \cdot 10^{n/2} \;+\; X_1Y_1$$

e aqui nós vemos que nós não precisamos dos resultados intermediários X_2Y_1 e X_1Y_2 separadamente, mas apenas da sua soma.

E agora, tudo o que é preciso é uma pequena esperteza ...

Veja o que acontece quando nós deixamos as potências de 10 de lado e multiplicamos

$$\big(X_2 + X_1\big) \cdot \big(Y_2 + Y_1\big) \;=\; X_2Y_2 \;+\; \big(X_2Y_1 \;+\; X_1Y_2\big) \;+\; X_1Y_1$$

Ou seja, o termo que nos interessa apareceu de novo.

E a observação importante aqui é que, se as somas do lado esquerdo são realizadas primeiro, então a coisa toda leva tempo $\frac{n}{2} + \frac{n}{2} + \frac{n^2}{4}$.

O único problema é que essa multiplicação nos dá o termo que nós desejamos misturado com

X_2Y_2 e X_1Y_1.

Mas, isso é um problema fácil de resolver

- basta calcular os termos X_2Y_2 e X_1Y_1 em separado
- e depois fazer a subtração

$$\big(X_2Y_1 \ + \ X_1Y_2\big) \ = \ X_2Y_2 \ + \ \big(X_2 + X_1\big) \cdot \big(Y_2 + Y_1\big) \ + \ X_1Y_1$$

Ora, mas os termos X_2Y_2 e X_1Y_1 já eram termos que eram calculados na solução original.

Portanto, não há perda de tempo (realmente) aqui.

Em resumo, a esperteza consiste em calcular apenas 3 substituições:

- X_2Y_2
- X_1Y_1
- $X_2Y_2 \ + \ \big(X_2 + X_1\big) \cdot \big(Y_2 + Y_1\big) \ + \ X_1Y_1$

(O procedimento também envolve algumas somas e subtrações, mas a sua contribuição para o tempo total não é relevante.)

Finalmente, uma vez que esses 3 termos foram calculados, a solução é obtida através de

$$X \cdot Y \ = \ X_2Y_2 \cdot 10^n \ + \ \big(X_2Y_1 \ + \ X_1Y_2\big) \cdot 10^{n/2} \ + \ X_1Y_1$$

Pseudo-código e análise de complexidade

Uma vez que as estratégias de divisão e conquista foram definidas, é imediato escrever o algoritmo que implementa essa solução:

```
Procedimento  Mult-intDC ( X,Y : n dígitos )
 {
     Se  ( X e Y são pequenos )  Retorna ( X * Y )

     (X2,X1)  <--  Quebra (X)
     (Y2,Y1)  <--  Quebra (Y)

          C1  <--  Mult-intDC (X2,Y2)
          C2  <--  Mult-intDC (X1,Y1)
          C3  <--  Mult-intDC (X2+X1,Y2+Y1)  -  C1  -  C2

           R  <--  C1 * 10^n  +  C3 * 10^n/2  +  C2
 }
```

Examinando o pseudo-código, nós obtemos a seguinte equação de recorrência para o tempo de execução do algoritmo

$$T(n) \;=\; 3 \cdot T(n/2) \;+\; O(n)$$

que possui solução

$$T(n) \;=\; \Theta\big(n^{\log_2 3}\big) \;=\; \Theta\big(n^{1{,}585}\big)$$

4 Multiplicação eficiente de matrizes

As ideias que acabamos de ver podem ser aplicadas de maneira muito semelhante ao problema da multiplicação de matrizes.

Considere duas matrizes X, Y com dimensões $ntimesn$, particionadas em 4 blocos do mesmo tamanho:

$$X \;=\; \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad\qquad Y \;=\; \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

Então, a multiplicação de X e Y pode ser expressa como

$$X \cdot Y \;=\; \begin{bmatrix} (AE + BG) & (AF + BH) \\ (CE + DG) & (CF + DH) \end{bmatrix}$$

Como no caso da multiplicação de inteiros, a solução não requer que os produtos AE, BG, etc. sejam calculados individualmente.

Tudo o que precisamos são as 4 somas acima.

Um dia, em que não tinha coisa melhor para fazer, V. Strassen ficou experimentando e descobriu que as somas podem ser calculadas realizando apenas 7 multiplicações.

A ideia é a seguinte.

Primeiro são calculados os seguintes componentes

$$\begin{aligned}
C_1 &= A \cdot \big(F \,-\, H\big) \\
C_2 &= \big(A \,+\, B\big) \cdot H \\
C_3 &= \big(C \,+\, D\big) \cdot E \\
C_4 &= D \cdot \big(G \,-\, E\big) \\
C_5 &= \big(C \,+\, D\big) \cdot \big(E \,+\, H\big) \\
C_6 &= \big(B \,-\, D\big) \cdot \big(G \,+\, H\big) \\
C_7 &= \big(A \,-\, G\big) \cdot \big(E \,+\, F\big)
\end{aligned}$$

A seguir, as somas são calculadas da seguinte maneira:

$$\begin{aligned} AE + BG &= C_5 + C_4 - C_2 + C_6 \\ AF + BH &= C_1 + C_2 \\ CE + DG &= C_3 + C_4 \\ CF + DH &= C_5 + C_1 - C_3 - C_7 \end{aligned}$$

Dessa maneira, nós obtemos um algoritmo de divisão e conquista para a multiplicação de matrizes com tempo de execução descrito pela recorrência

$$T(n) = 7 \cdot T(n/2) + O(n^2)$$

que possui solução

$$T(n) = \Theta\left(n^{\log_2 7}\right) = \Theta\left(n^{2,8}\right)$$

Seguindo o exemplo de Strassen, outros descobriram decomposições mais sofisticadas, e hoje já se conhecem algoritmos que realizam a multiplicação de matrizes em tempo $O(n^{2,37})$.