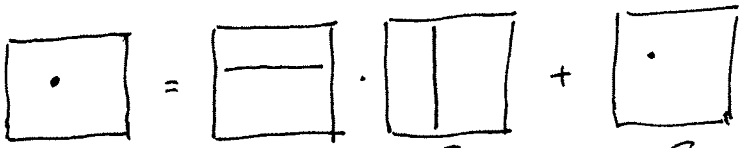
Matrix-matrix multiplication (MMM)



$C = A \cdot B + C$

square: $n \times n$ times $n \times n$

for $i = 0 \dots n-1$
  for $j = 0 \dots n-1$
    for $l = 0 \dots n-1$
      $c_{i,j} = c_{i,j} + a_{i,l} \cdot b_{l,j}$

rectangular: $k \times m$ times $m \times n$

for $i = 0 \dots k-1$
  for $j = 0 \dots n-1$
    for $l = 0 \dots m-1$
      $c_{i,j} = c_{i,j} + a_{i,l} \cdot b_{l,j}$

asymptotic runtime

$O(n^3)$ | $O(kmn)$

cost = (# adds, # mults)

(n^3, n^3) | (kmn, kmn)

cost = # flops

$2n^3$ | $2kmn$

Cost analysis: solving recurrences

First order: $f_0 = c, \quad f_k = a f_{k-1} + s_k, \quad k \geq 1$

$$\Rightarrow f_k = a^k c + \sum_{i=0}^{k-1} a^i s_{k-i}$$

Example: $f_0 = 0, \quad f_k = 2 \cdot f_{k-1} + 3 \cdot 2^{k-1} - 1$

$$\Rightarrow f_k = \sum_{i=0}^{k-1} 2^i (3 \cdot 2^{k-i-1} - 1) = \dots = \frac{3}{2} k 2^k - 2^k + 1$$

Exponential version: $g_1 = c, \quad g_n = a\, g_{n/2} + t_n \qquad n = 2^k$

[substitute $n = 2^k$, $g_n = f_k$, $t_n = s_k$]

$\Rightarrow f_0 = c, \quad f_k = a f_{k-1} + s_k$

solve as before, translate back

Example: $g_1 = 0, \quad g_n = 2 g_{n/2} + \frac{3}{2} n - 1$ $\xrightarrow{\text{Master theorem}}$ $g_n = \Theta(n \log n)$

$\overset{n = 2^k}{\Longrightarrow} f_0 = 0, \quad f_k = 2 f_{k-1} + 3 \cdot 2^{k-1} - 1$

$\overset{\text{solve}}{\Longrightarrow} f_k = \frac{3}{2} k 2^k - 2^k + 1$

$\overset{\text{translate}}{\Longrightarrow} g_n = \frac{3}{2} n \log_2(n) - n + 1$