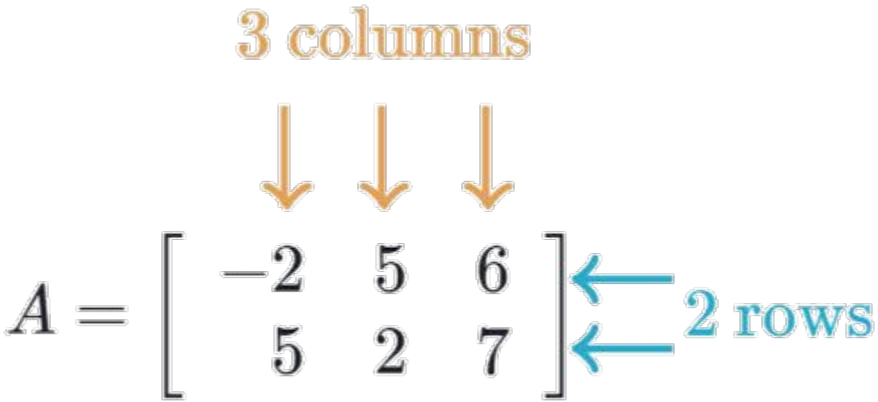
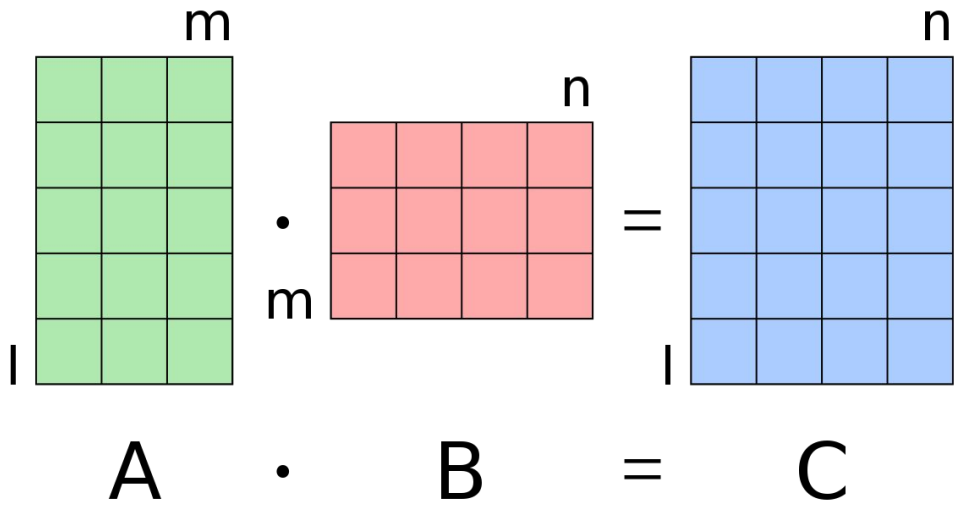
Review: Matrices

Matrix: 2D array of numbers (n rows, m columns → **n x m** matrix)



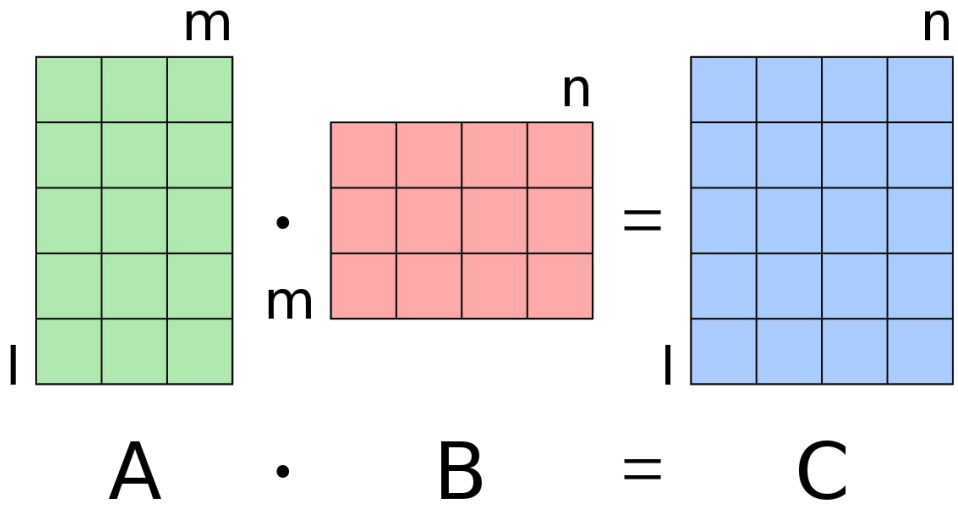
Matrix multiplication

To multiply matrices A and B, the second dimension of A must be the first dimension of B.



Matrix multiplication

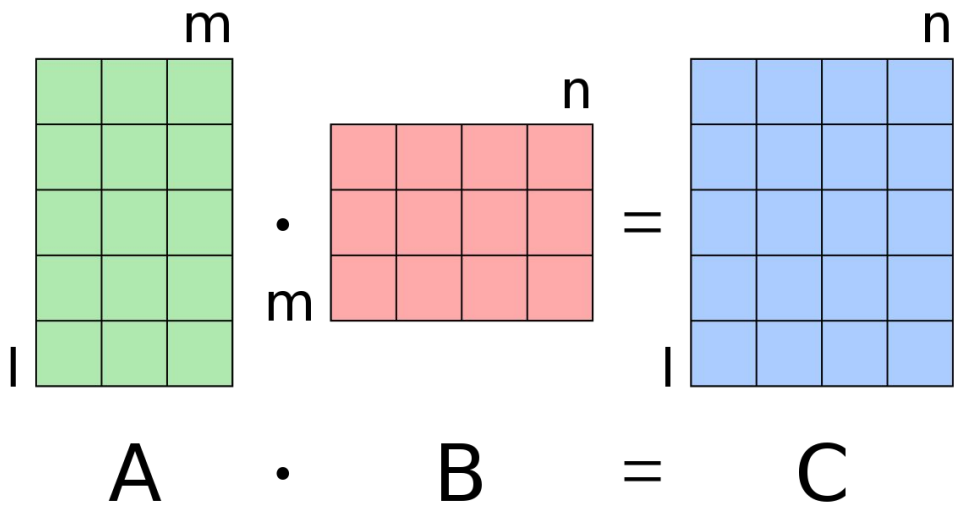
Usages: Many modern data-intensive processes, such as machine learning and dataset management, rely on matrix multiplication.



Matrix multiplication

Let C = A x B. The entry C[i][j] = sum(A[i][k] * B[k][j] for k = 0 … m-1).

- **dot product** of ith row of A, jth column of B



Matrix multiplication: Example

$$\begin{bmatrix} 1 & 8 \\ 3 & 6 \end{bmatrix} \times \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

Matrix multiplication: Example

$$\begin{bmatrix} 1 & 8 \\ 3 & 6 \end{bmatrix} \times \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 1*4 + 8*3 \\ 3*4 + 6*3 \end{bmatrix}$$

$$= \begin{bmatrix} 28 \\ 30 \end{bmatrix}$$

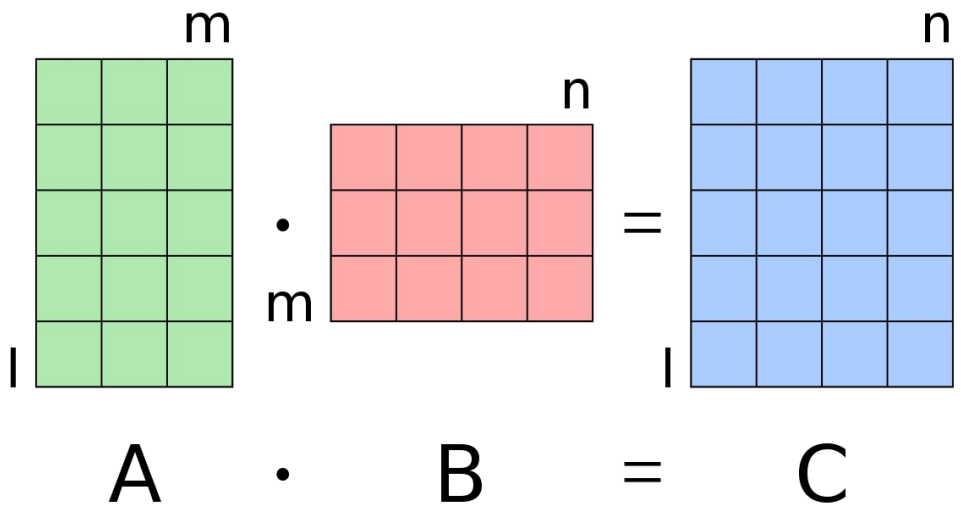
Time complexity of matrix multiplication

For each entry in C (l x n):

Need to compute the summation of m different product terms.

Therefore, overall complexity is **O(lmn)**.

For square matrices, is **$O(n^3)$**.



Matrix multiplication is slow…

Naive method: $\mathbf{O(n^3)}$

Strassen's method: $\mathbf{O(n^{\log 7}) \approx O(n^{2.8})}$

Williams et al (2020): $\mathbf{O(n^{2.3728596})}$

Duan et al (2022): $\mathbf{O(n^{2.371866})}$

Williams et al (2023): $\mathbf{O(n^{2.371552})}$